



Programação Estruturada

Professor Adjenor Cristiano
Queiroz
FAPAM - 2023

AULA 6



HORA DA
CORREÇÃO!



EXERCÍCIOS

1 - Desenvolva um software que imprima na tela a renomada canção da música popular brasileira Cinco patinhos da Xuxa. Utilize a estrutura de repetição ENQUANTO.

“5 patinhos foram passear

Além das montanhas para brincar

A mamãe gritou: Quá, quá, quá, quá

Mas só 4 patinhos voltaram de lá”

```
5 patinhos foram passear
Além das montanhas para brincar
A mamãe gritou: Quá, quá, quá, quá
Mas só 4 patinhos voltaram de lá
```

```
4 patinhos foram passear
Além das montanhas para brincar
A mamãe gritou: Quá, quá, quá, quá
Mas só 3 patinhos voltaram de lá
```

```
3 patinhos foram passear
Além das montanhas para brincar
A mamãe gritou: Quá, quá, quá, quá
Mas só 2 patinhos voltaram de lá
```

```
2 patinhos foram passear
Além das montanhas para brincar
A mamãe gritou: Quá, quá, quá, quá
Mas só 1 patinho voltou de lá
```

```
1 patinho foi passear
Além das montanhas para brincar
A mamãe gritou: Quá, quá, quá, quá
Mas nenhum patinho voltou de lá
```

```
1 Algoritmo "Musica-Patinhos"
2 Var
3 cont: inteiro
4 Inicio
5 cont <- 5
6 enquanto (cont > 0) faca
7     se (cont > 1) entao
8         escreval(cont, " patinhos foram passear")
9     senao
10        escreval(cont, " patinho foi passear")
11    fimse
12    escreval("Além das montanhas para brincar")
13    escreval("A mamãe gritou: Quá, quá, quá, quá")
14    cont <- (cont - 1)
15    se (cont > 1) entao
16        escreval("Mas só ", cont, " patinhos voltaram de lá")
17    senao
18        se (cont = 1) entao
19            escreval("Mas só ", cont, " patinho voltou de lá")
20        senao
21            escreval("Mas nenhum patinho voltou de lá")
22        fimse
23    fimse
24    escreval("")
25 fimenquanto
26 Fimalgoritmo
```



EXERCÍCIOS

2 - Refaça o seu software do exercício 1 de modo que se o usuário digitar um valor maior que 100 ou menor que 0 ele peça ao usuário que digite novamente um valor até que o valor digitado esteja no intervalo de 0 a 100.

“Desenvolva um software que receba uma idade do usuário e imprima na tela se ele é maior de idade ou menor de idade”

```
1 Algoritmo "Idade"
2 Var
3   idade : inteiro
4 Inicio
5
6   enquanto (VERDADEIRO) FACA
7     escreval("Qual sua Idade? ")
8     leia(idade)
9     se ((idade >= 0) e (idade <= 100)) entao
10      INTERROMPA
11    senao
12      escreval("Idade Inválida! Digite um valor entre 0 e 100!")
13    fimse
14  fimenquanto
15
16  se (idade >= 18) entao
17    escreval("Maior de Idade ")
18  senao
19    escreval("Menor de Idade ")
20  fimse
21
22 Fimalgoritmo
```

1 – Desenvolva um software em que receberá alguns números fornecidos pelo usuário, enquanto ele quiser continuar digitando números (a cada interação pergunte ao usuário se ele deseja continuar a soma), e ao final, exiba o resultado da soma dos números digitados na tela.


```
1 Algoritmo "SomaNumeros"
2 Var
3   soma,nun : inteiro
4   opcao : caractere
5 Inicio
6   soma<-0
7   enquanto (VERDADEIRO) FACA
8     escreval("Deseja somar mais algum número? [S/N]")
9     leia(opcao)
10    opcao<-MAIUSC(opcao)
11    se (opcao="N") entao
12      INTERROMPA
13    senao
14      escreval("Digite um número para somar:")
15      leia(nun)
16      soma<-soma+nun
17    fimse
18  fimenquanto
19  escreval("O Total da soma é: ",soma)
20 Fimalgoritmo
```

2 – Desenvolva um software em que receba uma quantidade do usuário e imprima na tela essa quantidade de vezes “Programação é nossa melhor matéria!”.

```
1 Algoritmo "MelhorMateria"
2 Var
3   qtd, cont : inteiro
4 Inicio
5   cont <- 0
6   escreval("Quantas vezes deseja escrever na Tela?")
7   leia(qtd)
8   enquanto (qtd > cont) FACA
9     escreval("Programação é nossa melhor matéria!")
10    cont <- cont + 1
11  fimenquanto
12 Fimalgoritmo
```

3 – Atualize o software do exercício da Lanchonete de modo que o usuário possa fazer um pedido de vários itens no mesmo pedido. Quando ele desejar finalizar o pedido o sistema irá imprimir na tela a lista de produtos comprados e o valor da conta.

Dicas:

- Trabalhe dentro de um laço de repetição enquanto.
- Utilize uma variável do tipo caractere para armazenar os dados do pedido e a cada novo item adicionado, concatene a variável com ela mesma.

```
1 Algoritmo "Lanchonete"
2
3 Var
4 totalConta : REAL
5 opcao: INTEIRO
6 produtos : CARACTER
7
8 Inicio
9 totalConta <- 0
10 produtos<- ""
11 opcao<-0
12 escreval("GOSTARIA DE PEDIR ALGUM PRODUTO?")
13
14 enquanto (opcao<>5) FACA
15
16     escreval("-----")
17     escreval("                CARDÁPIO:                ")
18     escreval("")
19     escreval("1 - COXINHA:           R$ 6,00")
20     escreval("2 - PASTEL:           R$ 5,00")
21     escreval("3 - EMPADA:           R$ 5,50")
22     escreval("4 - PÃO DE QUEIJO: R$ 4,50")
23     escreval("5 - ENCERRAR A COMPRA")
24     escreval("-----")
25     escreval("")
26     leia(opcao)
```

```
28 ESCOLHA (opcao)
29     CASO 1
30         totalConta <- totalConta+6
31         produtos <- produtos+"1 - COXINHA:          R$ 6,00 |"
32     CASO 2
33         totalConta <- totalConta+5
34         produtos <- produtos+"1 - PASTEL:           R$ 5,00 |"
35     CASO 3
36         totalConta <- totalConta+5.5
37         produtos <- produtos+"1 - EMPADA:           R$ 5,50 |"
38     CASO 4
39         totalConta <- totalConta+4.5
40         produtos <- produtos+"1 - PÃO DE QUEIJO: R$ 4,50 |"
41     CASO 5
42         INTERROMPA
43     OUTROCASO
44         escreval("PRODUTO INVÁLIDO! ESCOLHA UM PRDUTO:")
45 FIMESCOLHA
46 escreval("")
47 SE ((opcao>1)E(opcao<5)) ENTAO
48     escreval("ÓTIMA ESCOLHA!")
49     escreval("GOSTARIA DE PEDIR ALGO MAIS?")
50 FIMSE
51 FIMENQUANTO
```



```
53 escreval("")
54 escreval("")
55 escreval("RESUMO DO PEDIDO:")
56 escreval("")
57
58 SE (produtos="") ENTAO
59     escreval("NENHUM PRODUTO ADICIONADO!")
60 FIMSE
61
62 escreval(produtos)
63 escreval("TOTAL: R$ ", totalConta:4:2)
64
65 Fimalgoritmo |
```

OUTRAS

ESTRUTURAS DE REPETIÇÃO





REPITA ATÉ (SIMILAR AO DO WHILE - PHP, JAVA...)

Além do ENQUANTO, existem outras duas estruturas de repetição em Portugol: REPITA-ATÉ e PARA.

Podemos observar que na estrutura ENQUANTO o loop só vai acontecer se a condição for verdadeira quando for testada pela primeira vez, ou seja, os comandos que colocarmos no bloco do enquanto só vão ser executados se a expressão testada retornar um verdadeiro.



REPITA ATÉ (SIMILAR AO DO WHILE - PHP, JAVA...)

O funcionamento da estrutura REPITA-ATÉ é bem parecida com a do ENQUANTO, porém a condição é testada somente após o seu bloco de código ser executado.

Ou seja, os comandos são executados e depois é verificado se a expressão retorna um verdadeiro.

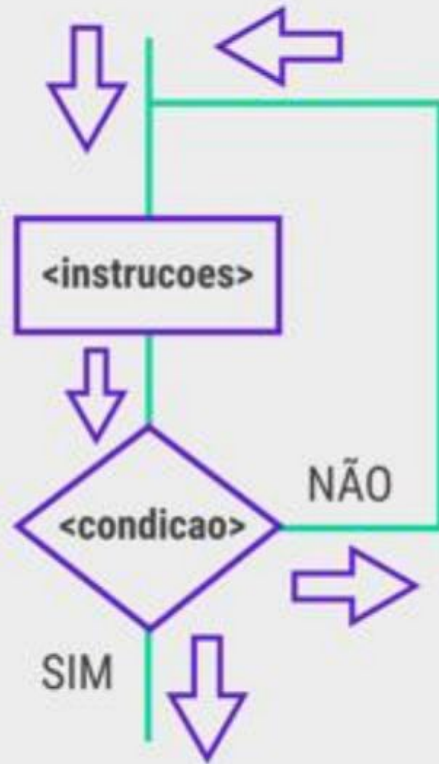


REPITA ATÉ

A primeira diferença entre o ENQUANTO e o REPITA-ATÉ é que, no REPITA-ATÉ, o loop vai ser executado ao menos uma vez, antes da condição que o controla ser testada, pois essa condição fica ao final da estrutura.

A principal diferença é que o REPITA-ATÉ interrompe a repetição se a condição de parada for verdadeira. Atualmente essa estrutura não existe na linguagem Python, embora possamos simular a suas funcionalidades.

REPITA ATÉ



REPITA ATÉ

Área dos algoritmos (Edição do código fonte) -> Nome do arquivo: [SOMA NUMEROS - REPITA-ATE.ALG]

```
1 algoritmo "SOMA NUMEROS"
2
3 var
4   resposta : caracter
5   numero, soma : INTEIRO
6 inicio
7
8   REPITA
9     ESCREVAL("Digite um Número para ser somado")
10    leia(numero)
11    soma<-numero + soma
12    ESCREVAL("Deseja continuar Somando? [S ou N]")
13    leia(resposta)
14  ATE ((resposta <> "S") E (resposta <> "s"))
15
16  ESCREVAL("SOMA: ",soma)
17
18
19 fimalgoritmo
20
```

ENQUANTO

Área dos programas (Edição do código fonte) -> Nome do arquivo: [SOMA NUMEROS - ENQUANTO.ALG]

```
1 algoritmo "SOMA NUMÉROS"
2
3 var
4   resposta : caracter
5   numero, soma : INTEIRO
6 inicio
7
8   resposta <- "S"
9   ENQUANTO ((resposta = "S") OU (resposta = "s")) FACA
10     ESCREVAL("Digite um Número para ser somado")
11     leia(numero)
12     soma <- numero + soma
13     ESCREVAL("Deseja continuar Somando? [S ou N]")
14     leia(resposta)
15   FIMENQUANTO
16
17   ESCREVAL("SOMA: ", soma)
18
19
20 fimalgoritmo
21
```

REPITA ATÉ

ENQUANTO

```
1 algoritmo "SOMA NUMEROS"
2
3 var
4     resposta : caracter
5     numero, soma : INTEIRO
6 inicio
7
8     REPITA
9         ESCREVAL("Digite um Número para ser somado")
10        leia(numero)
11        soma<-numero + soma
12        ESCREVAL("Deseja continuar Somando? [S ou N]")
13        leia(resposta)
14    ATE ((resposta <> "S") E (resposta <> "s"))
15
16    ESCREVAL("SOMA: ",soma)
17
18
19 fimalgoritmo
20
```

```
1 algoritmo "SOMA NUMÉROS"
2
3 var
4     resposta : caracter
5     numero, soma : INTEIRO
6 inicio
7
8     resposta<-"S"
9     ENQUANTO ((resposta = "S")OU(resposta = "s"))FACA
10        ESCREVAL("Digite um Número para ser somado")
11        leia(numero)
12        soma<-numero + soma
13        ESCREVAL("Deseja continuar Somando? [S ou N]")
14        leia(resposta)
15    FIMENQUANTO
16
17    ESCREVAL("SOMA: ",soma)
18
19
20 fimalgoritmo
```

PARA (SIMILAR AO FOR)

Quando queremos implementar um loop com um número pré-definido de iterações, outra opção é utilizar a estrutura de repetição: PARA. Vamos voltar com o exemplo do algoritmo para contar até 10:

```
7   contador <- 0
8   enquanto (contador < 100) faça
9       escreva("Serei um bom aluno a partir de agora")
10      contador <- contador + 1
11  fimenquanto
```

Para utilizar o enquanto, precisamos criar uma variável "contador" para que ela controle a condição de parada. Essa variável deve ser incrementada a cada repetição do loop.



PARA

O PARA serve justamente para facilitar esse tipo de iteração. Partindo do pressuposto que sabemos quantas vezes a estrutura tem que se repetir, podemos usar a estrutura PARA, sem que haja a necessidade de inicializar ou incrementar um contador, pois a própria estrutura fará isso implicitamente e automaticamente.



PARA

Por exemplo, o algoritmo que o Barth teria de imprimir na tela 100 vezes “Serei um bom aluno” poderia ser simplificado com o uso da estrutura PARA:

PARA

Área dos programas (Edição do código fonte) -> Nome do arquivo: [CASTIGO BARTH - PARA.ALG]

```
1 algoritmo "CASTIGO BARTH PARA"
2
3 var
4     contador : INTEIRO
5 inicio
6
7     PARA contador de 1 ATE 100 FACA
8         escreval("Serei um bom aluno a partir de agora")
9     FIMPARA
10
11 fimalgoritmo
```



PARA

Basicamente, podemos ler a estrutura acima como: repita o bloco de código de 1 até 100 (100 VEZES). Em cada repetição, a variável contador é incrementada em + 1, mudando, assim, o valor que estará sendo impresso

O interessante é que não precisamos implementar nenhum incrementador explicitamente, pois isso é feito automaticamente, quando usamos essa estrutura.



PARA - EXEMPLO PRÁTICO

Desenvolva um algoritmo de Tabuada de Multiplicação, onde o usuário digite um número e o software deve imprimir esse número multiplicado por números de 1 a 10.

Vamos utilizar o comando **PARA** e pedir ao usuário que digite o número que ele deseja a tabuada.

TABUADA PARA

Área dos programas (Edição do código fonte) -> Nome do arquivo: [CASTIGO BARTH - PARA.ALG]

```
1 algoritmo "TABUADA PARA"
2
3 var
4     cont, num : INTEIRO
5 inicio
6     escreva("Qual número você deseja a Tabuada? ")
7     leia(num)
8     PARA cont de 1 ATE 10 FACA
9         escreval(num, " X", cont, " =", (num*cont))
10    FIMPARA
11
12 fimalgoritmo
13
```

Portugol Studio

```
programa {  
  
    funcao inicio() {  
        inteiro numero, resultado, contador  
  
        escreva("Digite o para calcular o tabuada: ")  
        leia(numero)  
  
        para (contador = 1; contador <= 10; contador++) {  
            resultado = numero * contador  
            escreva (numero, " X ", contador, " = ", resultado , "\n")  
        }  
    }  
}
```



FINALIZAR UMA ESTRUTURA DE REPETIÇÃO

Pode haver casos em que precisamos finalizar nossa estrutura de repetição antes do momento predefinido pelo sistema ou pelo usuário. Para isso temos o comando **INTERROMPA**, que pode ser utilizado nas estruturas **PARA**, **ENQUANTO**, **REPITA** e **ESCOLHA**.

FINALIZAR UMA ESTRUTURA DE REPETIÇÃO

Área dos algoritmos (Edição do código fonte) -> Nome do arquivo: [CASTIGO BARTH PARA INTERRO

```
1 algoritmo "CASTIGO BARTH PARA"
2
3 var
4     qtd, contador : INTEIRO
5 inicio
6
7     escreval("QUANTAS IMPRESSÕES?")
8     leia(qtd)
9     PARA contador de 1 ATE qtd FAÇA
10         escreval(contador," - Serei um bom aluno!")
11
12         se (contador = 60) entao
13             INTERROMPA
14         fimse
15
16     FIMPARA
17
18 fimalgoritmo
19
```




EXERCÍCIOS - PARA

- 1 - Escreva um algoritmo para imprimir os 10 primeiros números inteiros maiores que 100.
- 2 - Escreva um algoritmo que imprima os números pares de 1 a 100. Utilize o sinal de % para verificar o resto da divisão.



EXERCÍCIOS

3 - Escreva um algoritmo que pergunte ao usuário quantos números ele quer imprimir na tela. Se o usuário digitar um número maior que 30, interrompa o laço de repetição e imprima na tela uma mensagem para o usuário.

4 - Escreva um algoritmo receba um número do usuário e imprima na tela os próximos 10 números múltiplos de 5. Utilize o comando INTERROMPA para sair do loop.



EXERCÍCIOS

5 - Escreva um algoritmo receba do usuário um número, depois receba do usuário uma opção (P para Par ou I para Ímpar) e imprima na tela todos os números de 0 até o número escolhido respeitando a opção escolhida (par ou ímpar). Ex: se o usuário digitar 6 e escolher ímpar, o sistema deve imprimir: **1 - 3 - 5**

6 - Reescreva o código do exercício anterior de modo que a impressão dos números seja escrita em ordem decrescente. Ex: se o usuário digitar 6 e escolher ímpar, o sistema deve imprimir: **5 - 3 - 1.**



EXERCÍCIOS

7 - Escreva um algoritmo para pedidos de uma loja de calçados. O algoritmo deve mostrar um menu com ao menos 4 produtos predefinidos e oferecer ao usuário a opção de selecionar mais de um produto ou encerrar o pedido (similar ao exercício do restaurante). Ao final o algoritmo deve mostrar o resumo da compra e o valor total (formatado em reais).



Bibliografia

- MENEZES, Nilo Ney Coutinho - Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes, 3ª Edição – 2019, Editora: Novatec Editora, ISBN-10: 8575227181
- SHAW, Zed A – Aprenda Python 3 do jeito certo, 1ª Edição – 2019, Editora: Alta Books, ISBN: 978-85-508-0473-6.
- <https://didatica.tech/a-linguagem-python/>
- <https://rockcontent.com/br/blog/algoritmo/>
- <https://sites.google.com/site/cleytoncaetanodesouza/disciplinas/algoritmos-e-l%C3%B3gica-de-programa%C3%A7%C3%A3o-portugol?pli=1>
- <https://edu.gcfglobal.org/pt/informatica-avancada/codigo-binario/1/>
-