

# Trilha Algoritmo

Encontro 03 - Estrutura de Repetição.



# Recapitulação

1. Estrutura condicional: Simples
2. Estrutura condicional: Composta
3. Estrutura condicional: Aninhada
4. Estrutura condicional: Escolha-Caso
5. Exemplos.
6. Atividades.

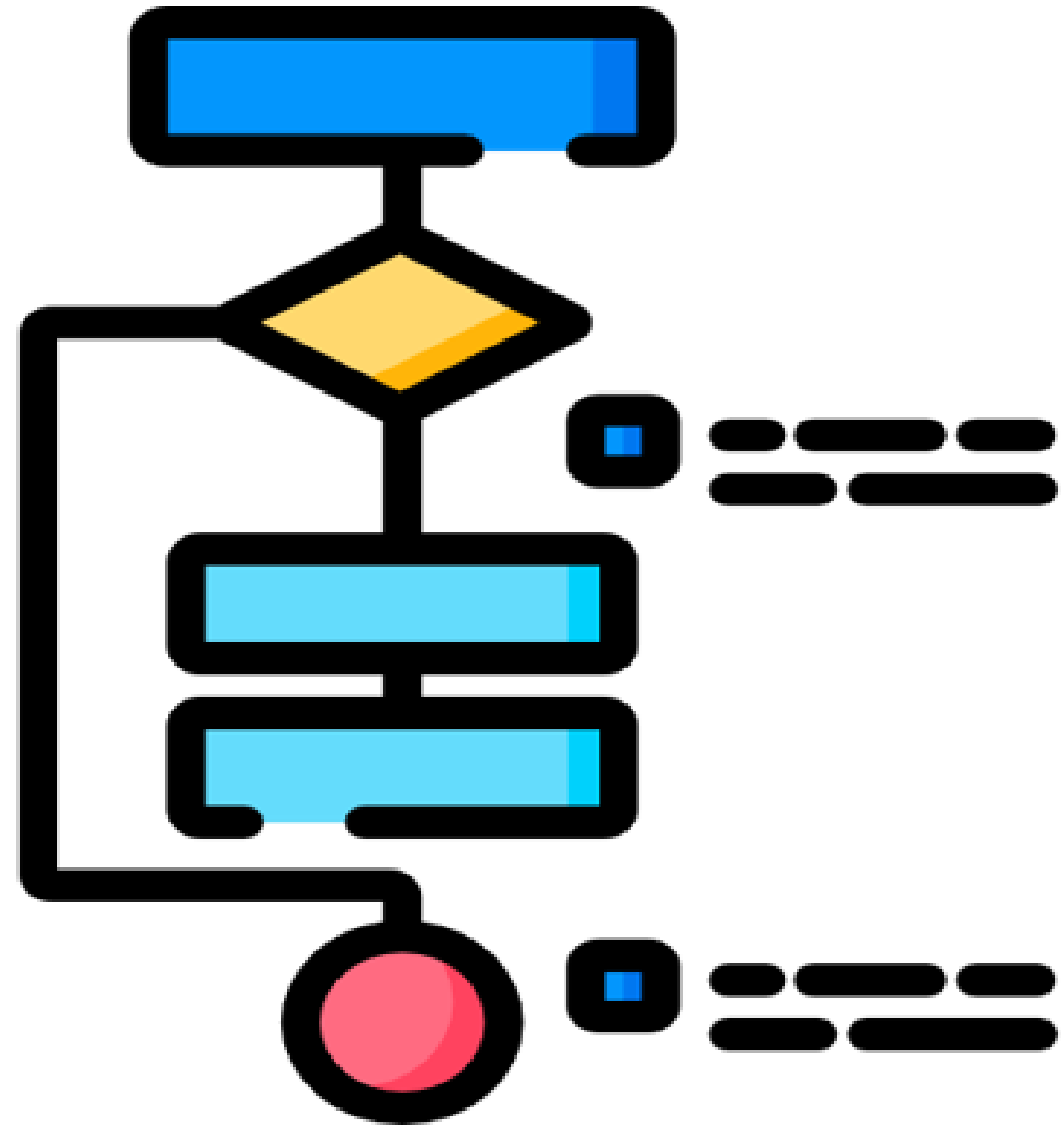


# Agenda

1. Estrutura de repetição: **repita...ate**
2. Estrutura de repetição: **enquanto...faca**
3. Estrutura de repetição: **para...faca**
4. Exemplos.
5. Atividades.



# Estrutura de Repetição

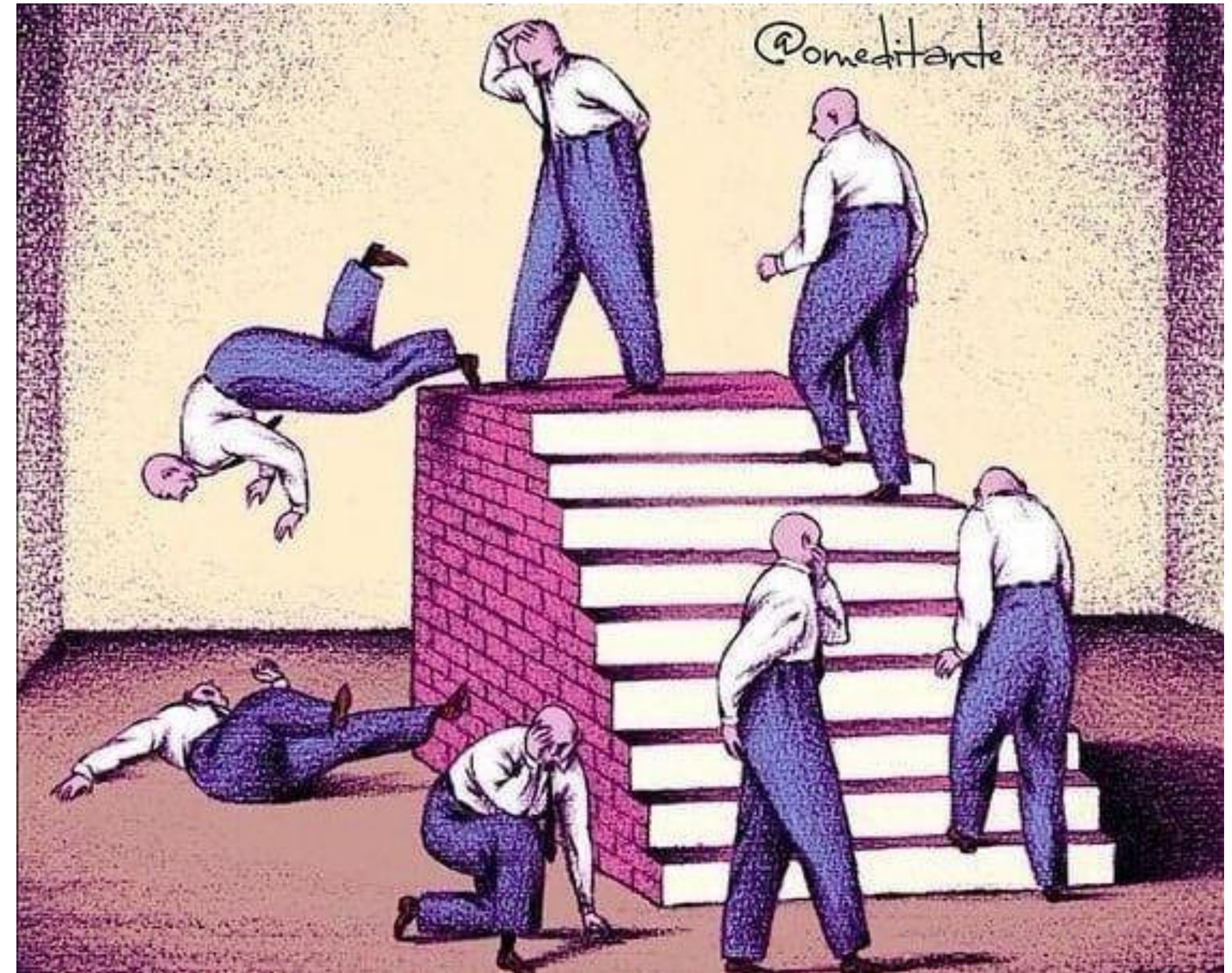




# Estrutura de Repetição

Até o momento foram **resolvidos** os problemas com uma **sequência de instruções** onde todas eram necessariamente executadas **uma única vez**.

Os **algoritmos** escritos seguiam, portanto, apenas uma **sequência linear** de operações.





# Estrutura de Repetição

Imagine desenvolver um algoritmo para ler a média das notas de 3 provas de um aluno.

Imagine ter que realizar esse processo por 40 vezes, já que numa dada sala de aula existe essa quantidade de aluno.

Existem partes do código que se repetem.



# Estrutura de Repetição

Este código  
não é sustentável

```
1 Algoritmo "MediaHarmonica"  
2 VAR  
3 a, b, c, MH: REAL  
4 NOME: caractere  
5 inicio  
6 escreva ("Entre com o nome do aluno: ")  
7 leia (nome)  
8 escreval ("Entre com as notas das três provas")  
9 escreva ("Digite a primeira nota: ")  
10 leia (a)  
11 escreva ("Digite a segunda nota: ")  
12 leia (b)  
13 escreva ("Digite a terceira nota: ")  
14 leia (c)  
15 MH <- 3/(1/a + 1/b +1/c)  
16 escreval ("A média harmônica do aluno: ", NOME, " é ", MH)  
17 escreva ("Entre com o nome do aluno: ")  
18 leia (nome)  
19 escreval ("Entre com as notas das três provas")  
20 escreva ("Digite a primeira nota: ")  
21 leia (a)  
22 escreva ("Digite a segunda nota: ")  
23 leia (b)  
24 escreva ("Digite a terceira nota: ")  
25 leia (c)  
26 MH <- 3/(1/a + 1/b +1/c)  
27 escreval ("A média harmônica do aluno: ", NOME, " é ", MH)
```



# Estrutura de Repetição

A solução anterior é viável apenas para uma turma de poucos alunos.

Para uma turma de 40 alunos, a codificação da solução seria por demais trabalhosa.

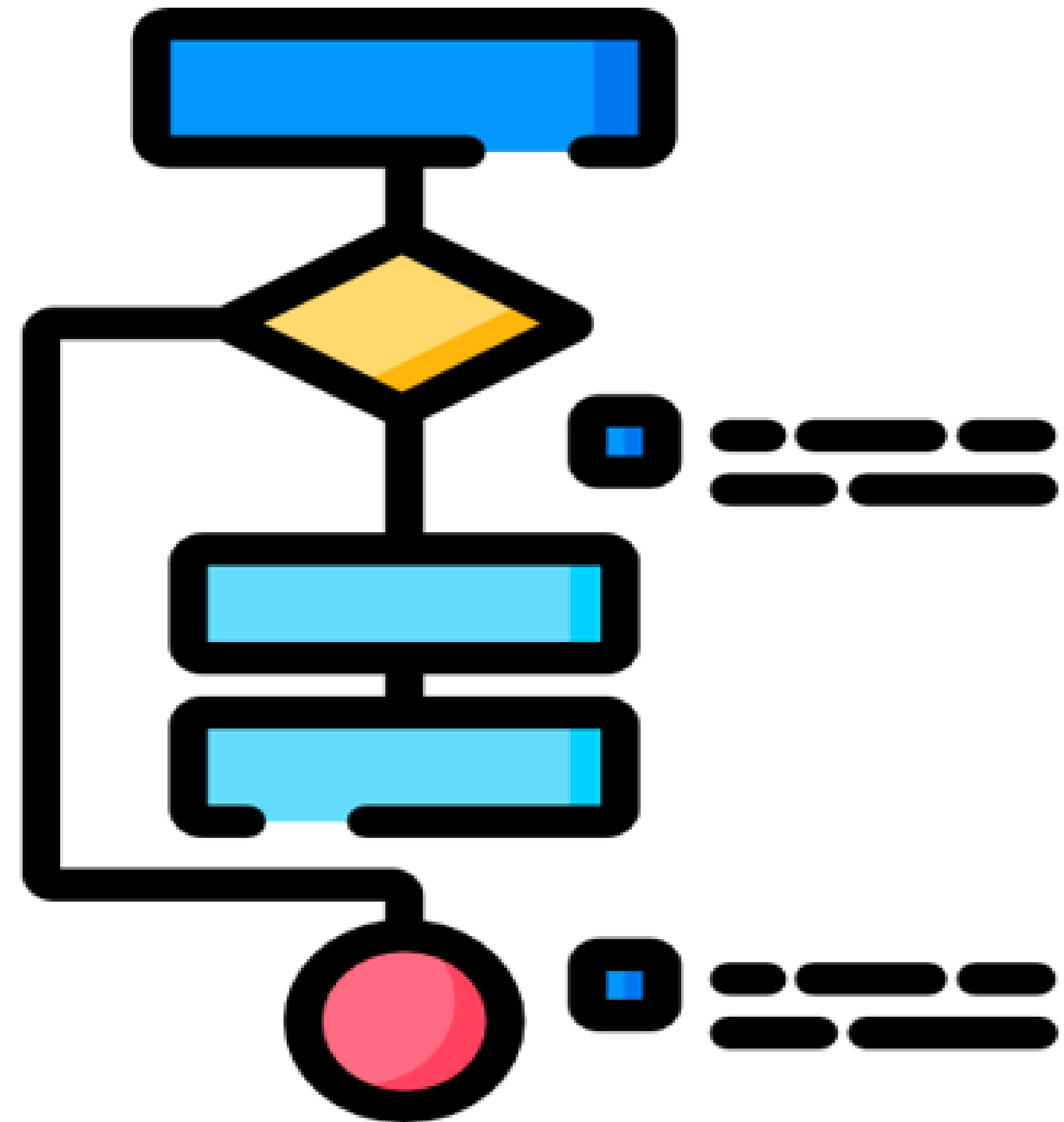
Existe um conjunto de estruturas de repetição que permite um algoritmo mais limpo e prático.

**Repita...ate / enquanto...faca / para...faca**



# Estrutura de Repetição

Repita...Ate



# Estrutura de Repetição

Todos os comandos da lista são executados e uma expressão lógica é avaliada.

Isto se repete até que a avaliação da condição resulte em VERDADEIRO

**repita**

<lista de comandos>

**ate** <expressão lógica>



# Estrutura de Repetição

Suponha um Algoritmo  
que escreve os  
números de 1 a 10.

```
1 algoritmo "Repeticao1"  
2  
3 VAR  
4   i: INTEIRO  
5  
6 inicio  
7   i ← 1  
8   repita  
9       escreva (i)  
10      i ← i + 1  
11  ate i > 10  
12 fimalgoritmo
```

# Estrutura de Repetição

A **variável** de controle do laço recebe um **valor inicial**, é **incrementada** (ou decrementada) de um valor constante no laço e tem seu valor **testado no final** do laço.

A **inicialização** da variável contadora deve acontecer **fora do laço**, antes do seu início.



# Estrutura de Repetição

O Algoritmo da média harmônica dos alunos pode ser reescrito de uma forma mais prática.

```
1 Algoritmo "MediaHarmonicaFinal"
2 var
3   a, b, c, MH, i: real
4   NOME: caractere
5 inicio
6   i <- 1
7   Repita
8     escreva ("Entre com o nome do aluno: ")
9     leia (nome)
10    escreval ("Entre com as notas das três provas")
11    escreva ("Digite a primeira nota: ")
12    leia (a)
13    escreva ("Digite a segunda nota: ")
14    leia (b)
15    escreva ("Digite a terceira nota: ")
16    leia (c)
17    MH <- 3 / (1/a + 1/b + 1/c)
18    escreval ("A média harmônica do aluno: ", NOME, " é ", MH)
19    i <- i + 1
20  ate i > 3
21 FimAlgoritmo
```

# Estrutura de Repetição

Existem diversas maneiras de implementar o mesmo laço, mas todo **laço** com variável de controle deve conter:

- a) inicialização da variável de controle;
- b) incremento (aumento do valor da variável de controle) ou decremento (diminuição do valor da variável de controle) da variável de controle; e
- c) teste de valor da variável de controle.



# Estrutura de Repetição

## Problema:

Faça um Algoritmo que escreve os números pares de 10 a 2. Use o comando Repita..até.

```
1 algoritmo "DecrementoNumerosPares"  
2 var  
3     i: inteiro  
4 inicio  
5     i <- 10  
6     Repita  
7         escreva (i)  
8         i <- i - 2  
9     ate i = 0  
10 Fimalgoritmo
```

# Estrutura de Repetição

## Um cuidado fundamental:

Certificar-se que a condição para que sejam mantidas as iterações torne-se, em algum momento, verdadeira.

Dessa forma, o algoritmo não entra em um laço infinito.





# Estrutura de Repetição

Qual o resultado desse algoritmo?

## Laço Infinito:

A inicialização da variável Contador deveria ser feita antes do comando repita.

```
1 algoritmo "laçoInfinito"  
2 VAR  
3     Contador: numerico  
4 inicio  
5     repita  
6         Contador <- 1  
7         Contador <- Contador + 1  
8     ate Contador = 10  
9 fimalgoritmo
```

# Estrutura de Repetição

Qual o resultado desse algoritmo?

## Laço Infinito:

A variável é incrementada de 2 em 2. Nunca será igual a 10.

```
1 algoritmo "laçoInfinito2"  
2 VAR  
3     Soma: numerico  
4 inicio  
5     Soma <- 1  
6     Repita  
7         Soma <- Soma + 2  
8     ate Soma = 10  
9     escreva (soma)  
10 fimalgoritmo
```

# Estrutura de Repetição

## Como corrigir o problema?

bastaria alterar a condição conforme o exemplo ao lado.

```
1 algoritmo "laçoInfinitoCorrigido"  
2 VAR  
3     Soma: REAL  
4 inicio  
5     Soma <- 1  
6     repita  
7         Soma <- Soma + 2  
8     ate Soma > 10  
9     escreva (soma)  
10 fimalgoritmo
```





Vamos  
praticar?

# Estrutura de Repetição

Escreva um algoritmo que leia uma quantidade de números e imprima o resultado informando a quantidade de números digitados que são negativos.

```
1 Algoritmo "repita3"  
2  
3 Var  
4     contador, numero, negativo: inteiro  
5  
6 Inicio  
7     contador <- 1  
8     negativo <- 0  
9     repita  
10        escreva("Digite um número: ")  
11        leia(numero)  
12        se (numero < 0) entao  
13            negativo <- negativo + 1  
14        fimse  
15        contador <- contador + 1  
16    ate(contador > 5)  
17    escreva("A quantidade de números negativos é: ", negativo)  
18 Fimalgoritmo
```

# Estrutura de Repetição

Escreva um algoritmo que leia um dado número e imprima na tela a sua tabuada.

```
1 Algoritmo "repita2"  
2  
3 Var  
4     contador, numero, soma: inteiro  
5  
6 Inicio  
7     contador <- 1  
8     escreva("Quer fazer a tabuada de qual número: ")  
9     leia(numero)  
10    repita  
11        soma <- contador*numero  
12        escreval(numero, " x", contador, " =", soma)  
13        contador <- contador + 1  
14    ate(contador > 10)  
15 Fimalgoritmo
```



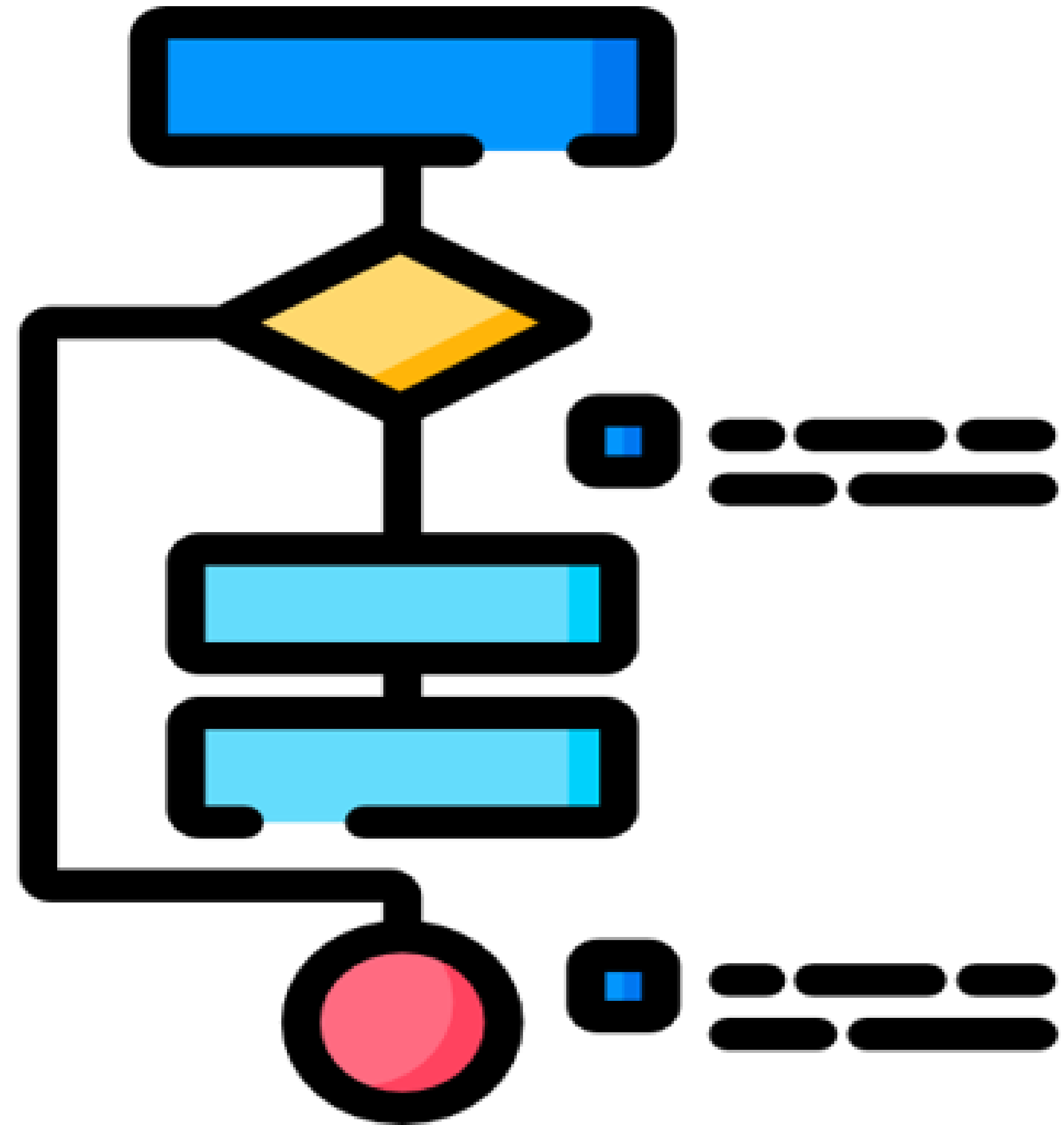
# Estrutura de Repetição

Suponha o mesmo algoritmo do exercício anterior, mas agora você deseja deixar a quantidade de números a ser digitado em aberto, ou seja, o usuário escolhe quantos números ele quer digitar. Como melhorar o algoritmo?

```
1 Algoritmo "repita4"  
2  
3 Var  
4   contador, numero, negativo, i: inteiro  
5  
6 Inicio  
7   contador <- 1  
8   negativo <- 0  
9   escreva("Quantos números você deseja digitar? ")  
10  leia(i)  
11  repita  
12    escreva("Digite um número: ")  
13    leia(numero)  
14    se (numero < 0) entao  
15      negativo <- negativo + 1  
16    fimse  
17    contador <- contador + 1  
18  ate(contador > i)  
19  escreva("A quantidade de números negativos é: ", negativo)  
20 Fimalgoritmo
```

# Estrutura de Repetição

Enquanto...faca



# Estrutura de Repetição

A expressão lógica é avaliada e, se ela for verdadeira, a lista de comandos é executada.

Isso se repete até que a condição seja falsa.

```
enquanto <expressão Lógica> faca  
    <lista de comandos>  
fimenquanto
```



# Estrutura de Repetição

A **diferença básica** entre as duas estruturas é a **posição onde é testada a expressão**.

Na **estrutura repita**, a condição é avaliada **após a execução dos comandos**, o que garante que os comandos serão executados pelo menos uma vez.

Na **estrutura enquanto**, a expressão é avaliada **no início** e se o resultado for FALSO no primeiro teste, a lista de comandos não é executada nenhuma vez.

# Estrutura de Repetição

## Exemplo:

Suponha um Algoritmo que conte de 1 até 10. Escreva uma frase quando terminar de contar.

```
1 Algoritmo "contador1"  
2  
3 Var  
4     contador: inteiro  
5  
6 Inicio  
7     contador <- 0  
8     enquanto (contador <=10) faca  
9         escreval(contador)  
10        contador <- contador+1  
11    fimenquanto  
12    escreva("Terminei de contar!!")  
13 Fimalgoritmo
```

# Estrutura de Repetição

## Exemplo:

Suponha um Algoritmo que conte de 10 até 0. Escreva uma frase quando terminar de contar.

```
1 Algoritmo "contador2"  
2  
3 Var  
4 contador: inteiro  
5  
6 Inicio  
7 contador <- 10  
8 enquanto (contador >= 0) faca  
9 escreval(contador)  
10 contador <- contador-1  
11 fimenquanto  
12 escreval("Terminei de contar!!")  
13 Fimalgoritmo
```



# Estrutura de Repetição

## Exemplo:

Suponha um Algoritmo onde o usuário digita até qual o número ele gostaria de fazer a contagem. Além disso, que ele escolha como essa contagem deve ser feita, por exemplo, com saltos de 2 em 2, ou saltos de 1 em 1, ou etc..

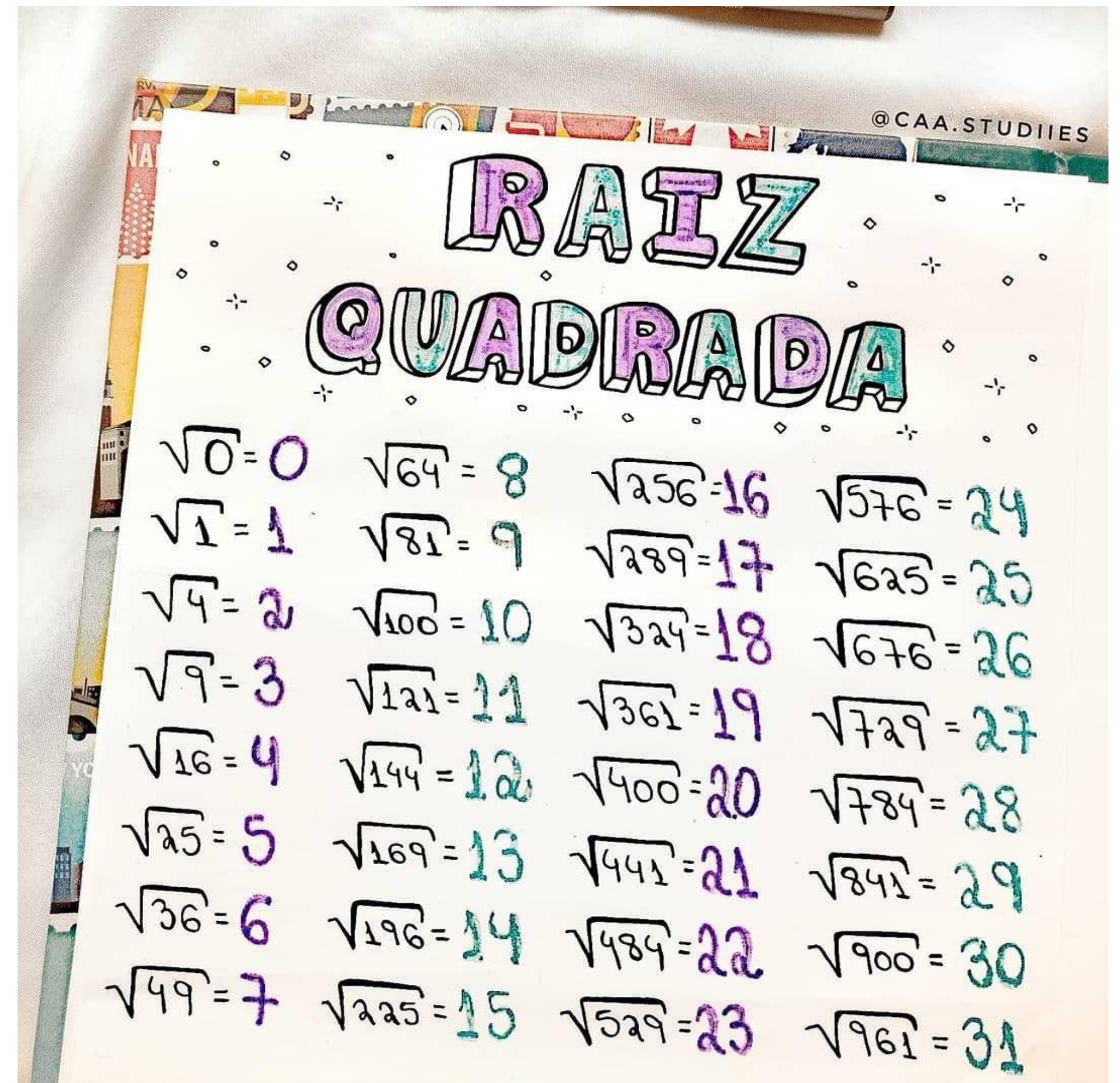
```
1 Algoritmo "contador3"
2
3 Var
4     contador, numero, salto: inteiro
5
6 Inicio
7     contador <- 0
8     escreva("Até qual número você quer contar? ")
9     leia(numero)
10    escreva("Voce quer contar saltando de quanto em quanto? ")
11    leia(salto)
12    enquanto (contador <= numero) faca
13        escreval(contador)
14        contador <- contador + salto
15    fimenquanto
16    escreval("Terminei de contar!!")
17 Fimalgoritmo
```



# Estrutura de Repetição

## Exemplo:

Suponha um Algoritmo que lê diversos números positivos e escreve, para cada um, sua raiz quadrada.





# Estrutura de Repetição

```
1 algoritmo "comEnquanto"  
2 var  
3   i: inteiro  
4 inicio  
5   leia (i)  
6   enquanto i >= 0 faca  
7     escreva (i^0.5)  
8     leia (i)  
9   fimenquanto  
10 fimalgoritmo
```

```
1 algoritmo "comRepita"  
2 var  
3   i: real  
4 inicio  
5   repita  
6     leia (i)  
7     se i >= 0 entao  
8       escreva (i^0.5)  
9     fimse  
10  ate i < 0  
11 fimalgoritmo
```





Vamos  
praticar?

# Estrutura de Repetição

## Exemplo:

Escreva um Algoritmo onde o usuário digita a quantidade de números que ele gostaria de somar. Faça com que o programa leia esses números e no final apresente o resultado da soma.

Use a estrutura **enquanto** **faca**.

```
1 Algoritmo "contador4"
2
3 Var
4     contador, soma, quantidade, numero: inteiro
5
6 Inicio
7     contador <- 1
8     soma <- 0
9     escreva("Quantos números você quer contar? ")
10    leia(quantidade)
11    enquanto (contador <= quantidade) faca
12        escreva("Digite o ", contador, ".o número: ")
13        leia(numero)
14        soma <- soma + numero
15        contador <- contador + 1
16    fimenquanto
17    escreva("A soma dos números é: ", soma)
18 Fimalgoritmo
```

# Estrutura de Repetição

Faça um Algoritmo que peça ao usuário a entrada de dois números, por exemplo, **num\_inicial** e **num\_final**. Assuma que a variável **i** recebe o valor de **num\_inicial** e enquanto ele for igual ou menor ao **num\_final** imprima o resultado. Em seguida imprima a palavra **FIM**.

Use a estrutura de repetição **enquanto..faça**



# Estrutura de Repetição

```
1 Algoritmo "numeros"
2
3 Var
4   num_inicial, num_final, i: inteiro
5 Inicio
6   Escreva("Digite o número inicial: ")
7   Leia(num_inicial)
8   Escreva("Digite o número final: ")
9   Leia(num_final)
10  i <- num_inicial
11  enquanto (i<=num_final) faça
12    Escreval(i)
13    i <- i+1
14  fimenquanto
15  Escreval("Fim!!")
16 Fimalgoritmo
```

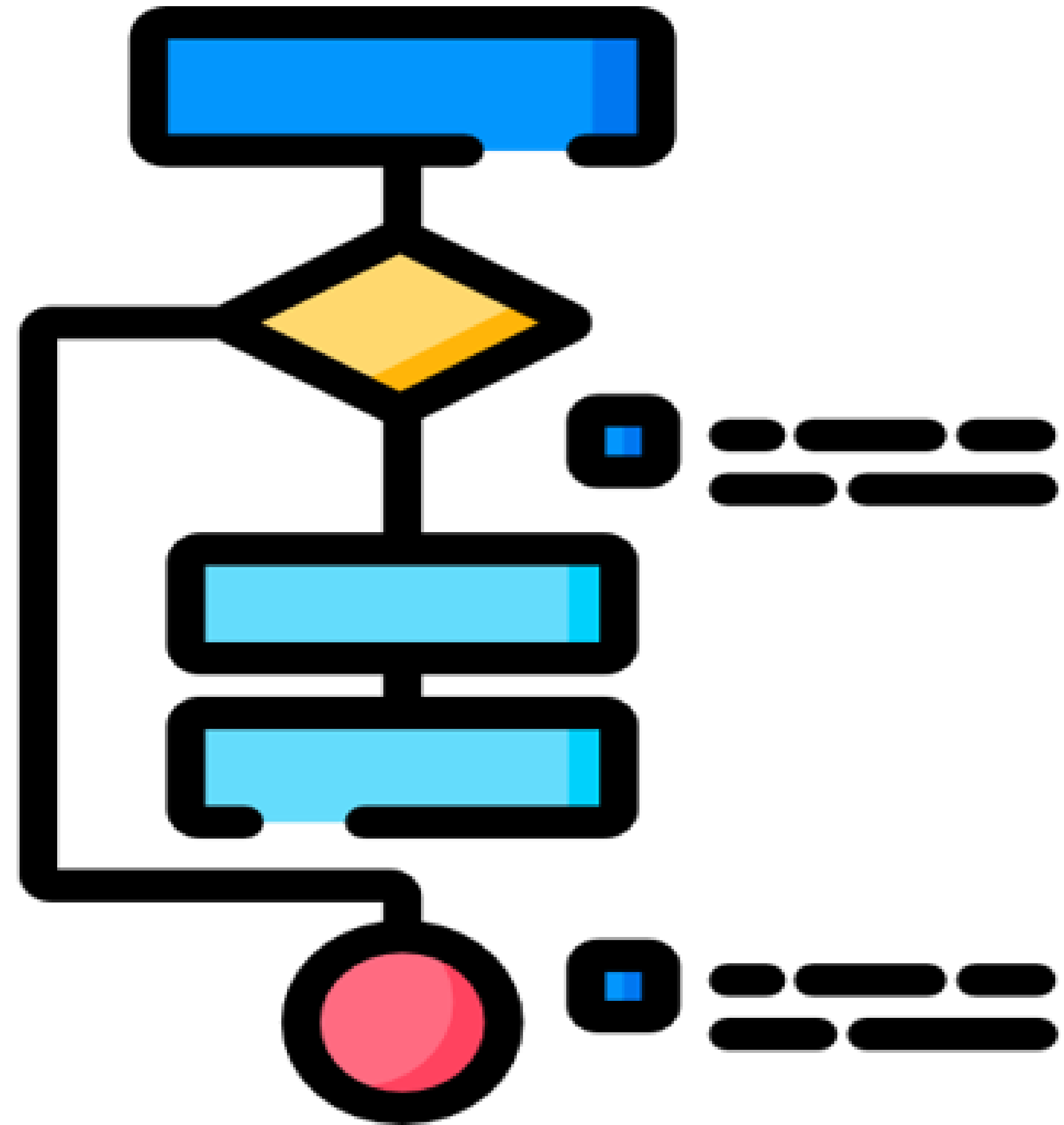


Coffee  
time!



# Estrutura de Repetição

Para...faca





# Estrutura de Repetição

O comando **para..faca** pode ser usado em algoritmos como uma estrutura de repetição.

```
Para <variável de controle> de <valor inicial> ate <valor final> [passo<incremento>] faca  
    <lista de comandos>  
fimpara
```

# Estrutura de Repetição

Suponha um algoritmo que lê e escreve os números ímpares de 1 a 100.

É possível usar a estrutura **para..faça**.

```
1 Algoritmo "impar"
2
3 Var
4 i:inteiro
5
6 Inicio
7     para i de 1 ate 100 passo 2 faça
8         escreva(i, " é ímpar")
9     fimpara
10 Fimalgoritmo
```

# Estrutura de Repetição

Suponha um Algoritmo que lê 5 números e imprima todos classificando-os como positivo ou negativo.

```
1 Algoritmo "PositivoNegativo"
2 var
3   i, numero: inteiro
4 inicio
5   para i de 1 ate 5 passo 1 faca
6     escreval ("Digite um numero")
7     leia (numero)
8     se numero > 0 entao
9       escreval (numero, " Positivo")
10    senão
11      escreval (numero, " Negativo")
12    fimse
13  fimpara
14 fimalgoritmo
```





Vamos  
Praticar!

# Estrutura de Repetição

## Exercício:

Escreva um Algoritmo que lê um número N e escreva todos os números de 1 até N.

```
1 algoritmo "tamanho do laço"
2 var
3     i, N: INTEIRO
4 inicio
5     leia (N)
6     para i de 1 ate N faca
7         escreva (i)
8     fimpara
9 fimalgoritmo
```

# Estrutura de Repetição

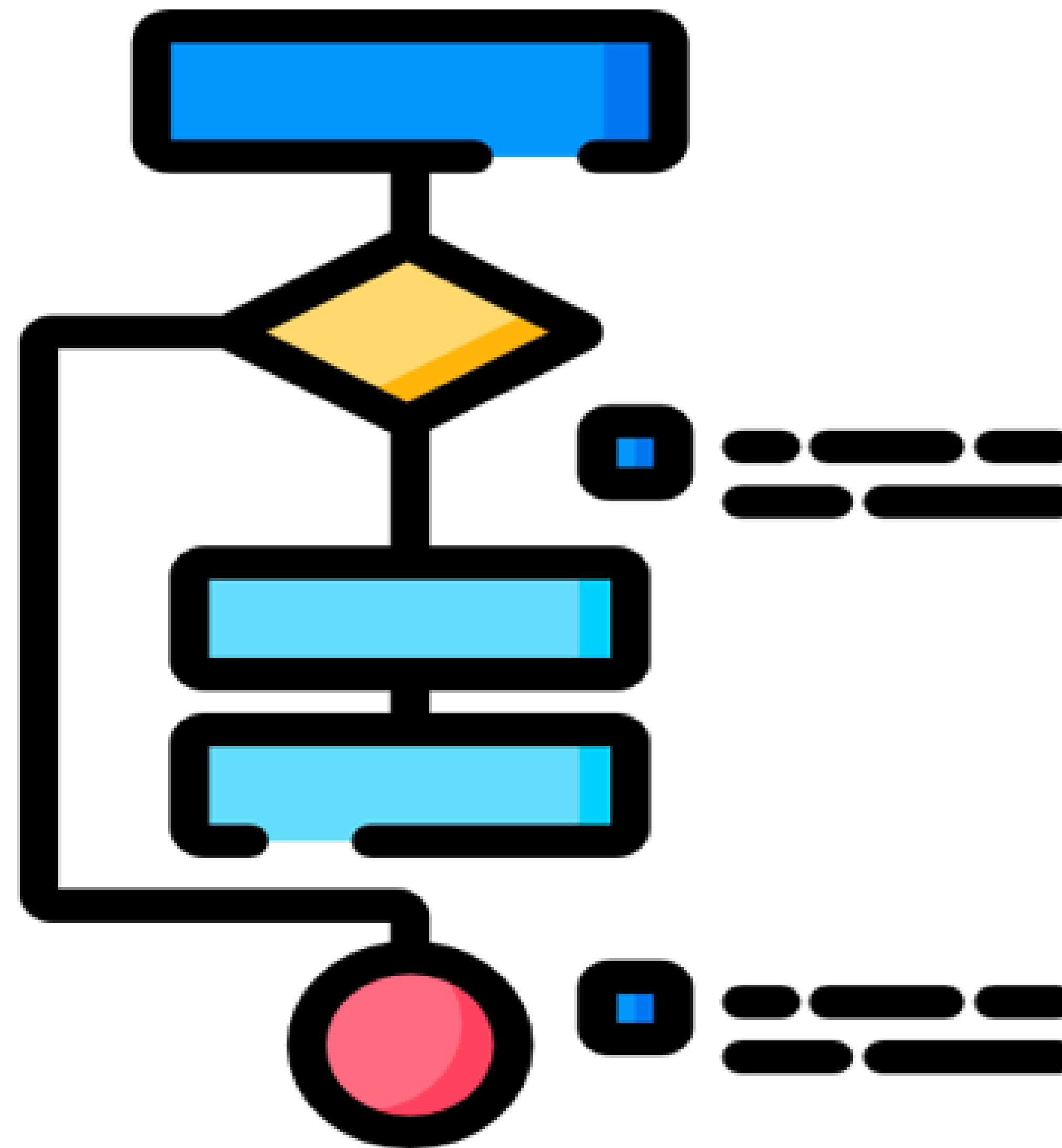
O valor  $I$  e  $N$  do exemplo acima tem que ser inteiro, pois se for declarado como um valor real ou numérico o algoritmo retornara com um erro de sitaxe.

Vale observar que, como nesse algoritmo é lido apenas um número, sua leitura deve ser feita fora da estrutura de repetição.



# Exercícios

## Estrutura de Repetição



# Prática

1) Faça um Algoritmo que calcule o fatorial de um número qualquer. O usuário deve digitar o número no qual ele deseja obter o resultado do fatorial na tela. Repita este processo até que o usuário não tenha mais interesse em obter cálculo de fatorial.

2) Faça um Algoritmo que leia um dado número e imprima na tela se este é um número primo ou não.

# Prática

3) Escreva um algoritmo que leia a nota final de 4 alunos. Suponha que Pedro e Maria tirem nota máxima 10, e que João e Sofia tirem a nota 9 e 8 respectivamente. Faça o programa identificar os alunos com a maior nota e imprima o resultado.

4) Escreva um algoritmo que gera e escreve os 4 primeiros números perfeitos. Um número perfeito é aquele que é igual à soma dos seus divisores. Ex:  $6 = 1+2+3$ ,  $28 = 1+2+4+7+14$ .



# Prática

5) Escreva um algoritmo que leia informações sobre um grupo de 250 pessoas e calcule alguns dados estatísticos.

Para cada pessoa do grupo deve leia o nome da pessoa, a altura, o peso e o sexo (“F” para feminino e “M” para o masculino).

Calcule e escreva: A quantidade total de homens e mulheres e o percentual de cada. A média de peso das pessoas (somatório dos pesos de todas as pessoas pela quantidade de pessoas).

O nome da pessoa mais alta.





# Prática

6) Faça um algoritmo que gere uma contagem inteligente. O programa deve ler dois números (inicial e final). Caso o número inicial seja maior que o final, escreva os números intermediários em ordem decrescente. Mas se o número inicial for menor que o final, escreva os números intermediários em ordem crescente. Imprima o resultado.





# Prática

7) Um funcionário de uma empresa recebe, anualmente, aumento salarial. Sabe-se que: a) esse funcionário foi contratado em 2016, com salário inicial de R\$2.000,00; b) Em 2017, ele recebeu aumento de 1,5%, sobre o seu salário inicial; c) A partir de 2018 (inclusive), os aumentos salariais sempre corresponderam ao dobro do percentual do ano anterior. Faça um programa que determine o salário desse funcionário dos anos de 2016 ao ano atual(2022). Apresente todos os valores.





# Prática

*8) Um determinado material radioativo perde metade de sua massa a cada 50 segundos. Dada a massa inicial, em gramas, fazer um programa que determine o tempo necessário para que essa massa se torne menor que 0,5 grama. Escreva a massa inicial, a massa final e o tempo calculado em horas e minutos.*



# Desafio

*9) Uma Universidade deseja fazer um levantamento a respeito de seu vestibular. Para cada curso é fornecido o seguinte conjunto de valores: Um código do curso; Número de vagas; Número de candidatos do sexo masculino; Número de candidatos do sexo feminino.*

*Fazer um programa que:*

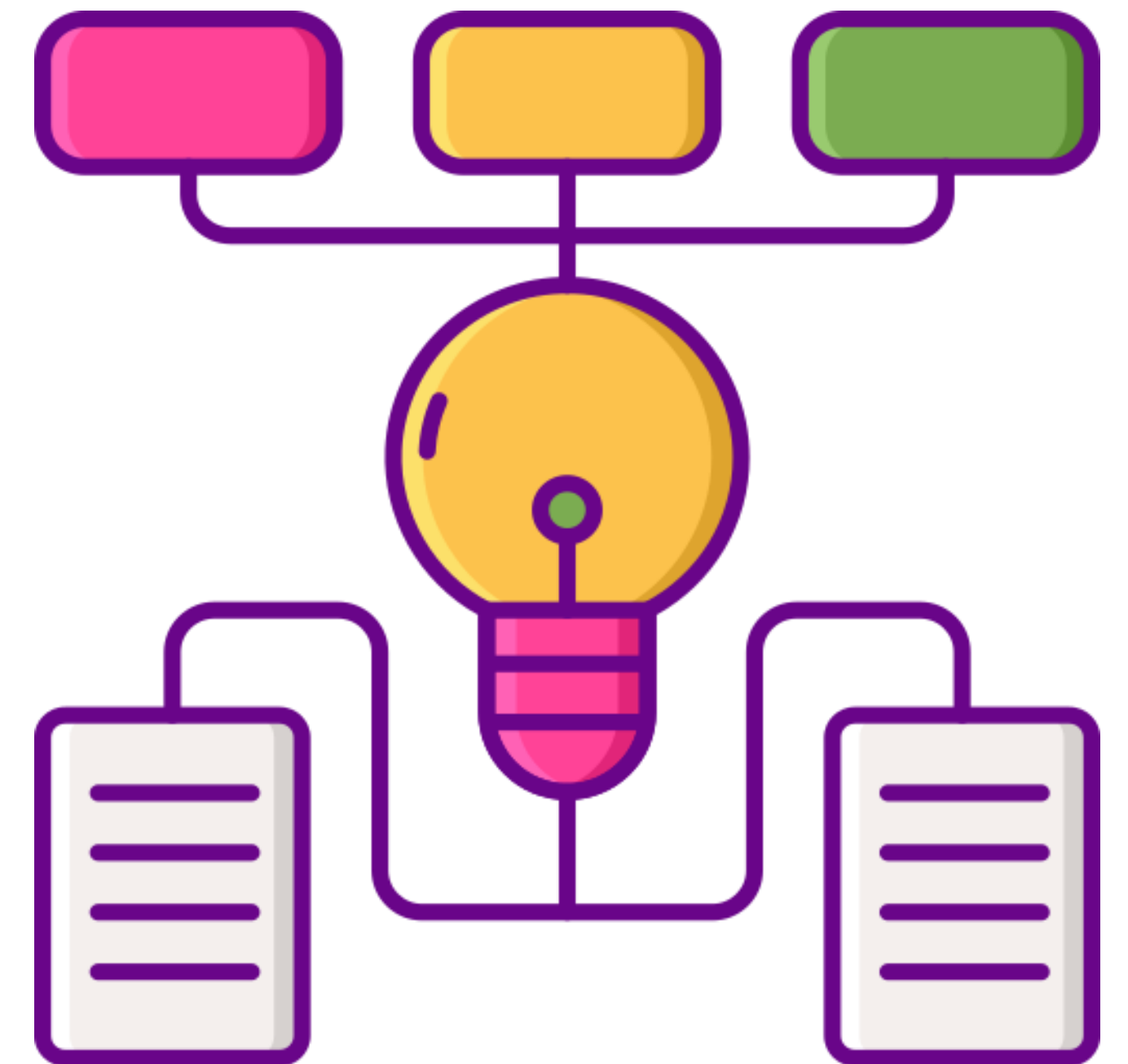
*Calcule e escreva, para cada curso, o número de candidatos por vaga e a percentagem de candidatos do sexo feminino (escreva também o código correspondente do curso);*



# Dica de hoje

## Tabelas Úteis

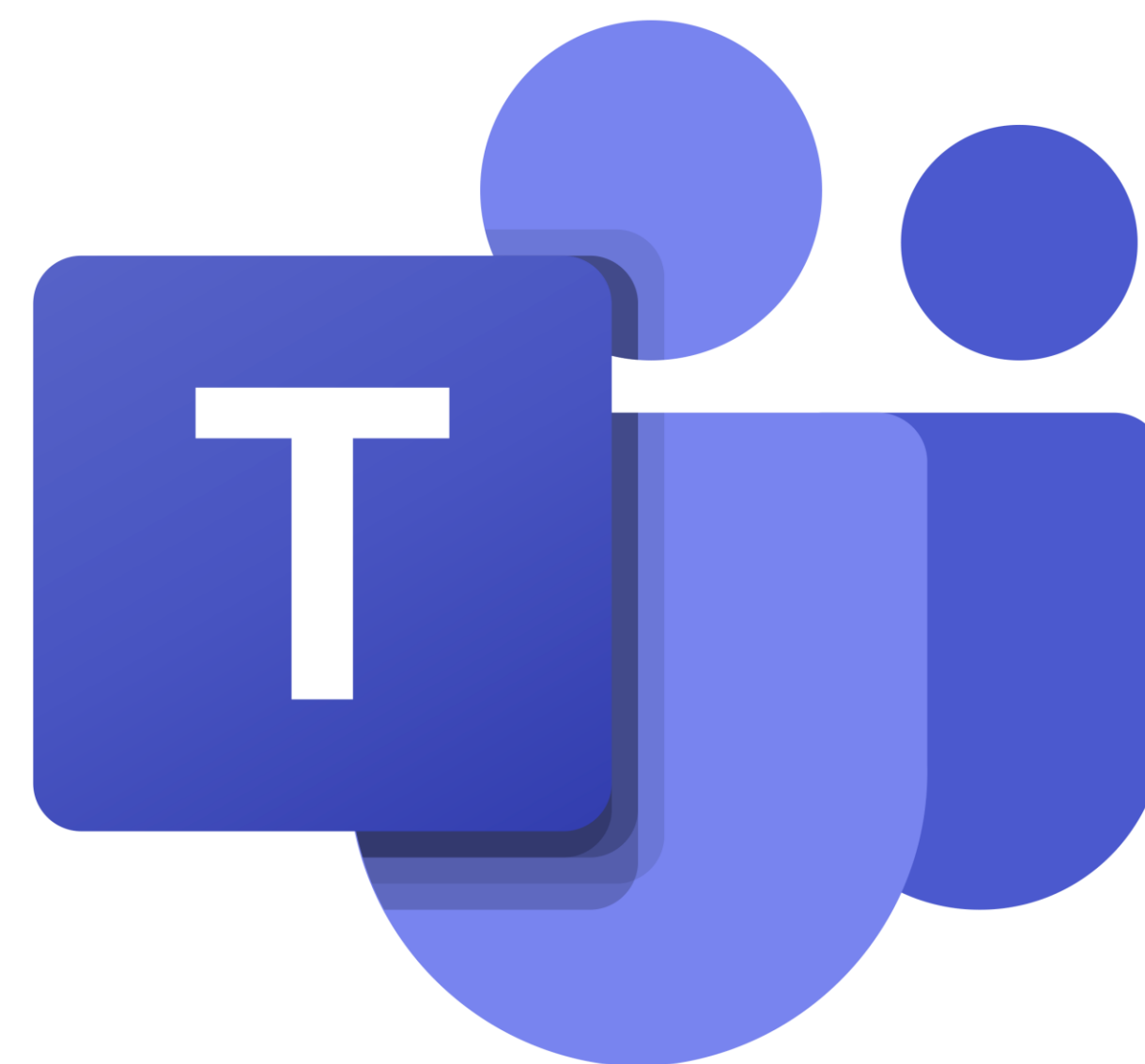
<https://visualg3.com.br/tabelas/>







# Comunidade VNT



# Referências

- [1] A. Goldman, F. Kon, Paulo J. S. Silva; Introdução à Ciência da Computação com Java e Orientação a Objetos (USP). 2006. Ed. USP.
- [2] Algoritmo e lógica de programação. Acessado julho/2022: <https://visualg3.com.br/>
- [3] G. Silveira; Algoritmos em Java; Ed. Casa do Código.
- [4] M. T. Goodrich, R. Tamassia; Estrutura de dados e algoritmos em Java. Ed Bookman. 2007.
- [5] Algoritmo e lógica de programação. Acessado julho/2022: <https://www.cursoemvideo.com/>
- [6] P. Silveira, R. Turini; Java 8 Prático: lambdas, streams e os novos recursos da linguagem. Ed. Casa do Código.
- [7] Linguagem Java: Curso acessado em agosto/2022: <https://www.udemy.com/>
- [8] Linguagem Java: Curso acessado em setembro/2022: <https://www.cursoemvideo.com/>

