

Trilha Algoritmo

Encontro 05 - Vetores e Matrizes



Recapitulação

1. O que é subprograma?
2. Procedimentos
3. Funções
4. Exemplos
5. Exercícios

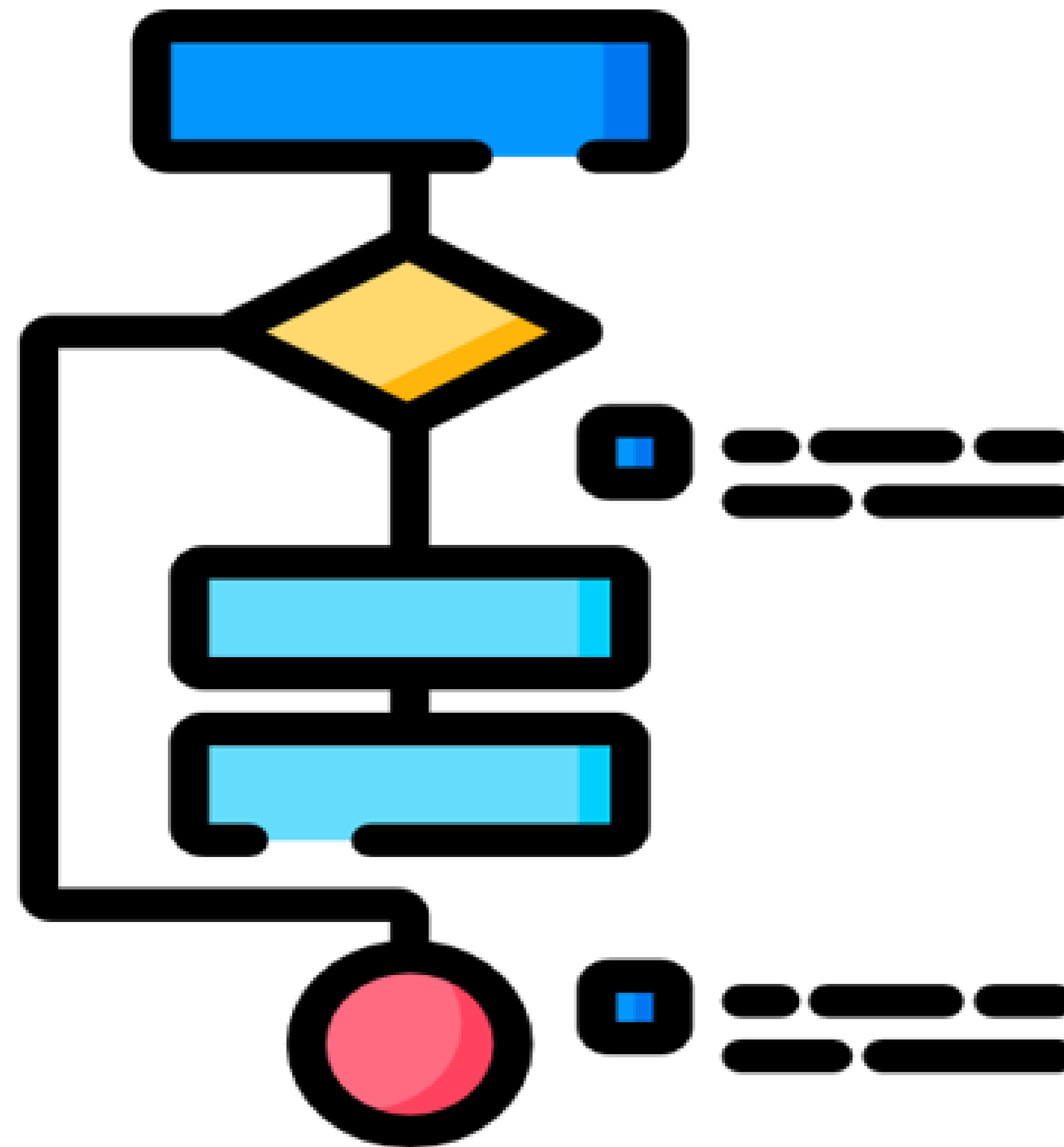


Agenda

1. Variáveis indexadas
2. Variáveis Indexadas Unidimensionais (**Vetores**)
3. Variáveis Indexadas Bidimensionais (**Matrizes**)
4. Exemplos
5. Atividades



Variáveis Indexadas



Variáveis Indexadas

A declaração de variáveis, uma a uma, é suficiente para a codificação algorítmica da solução de uma ampla gama de problemas.

Mas é insuficiente para resolver um grande número de problemas computacionais.

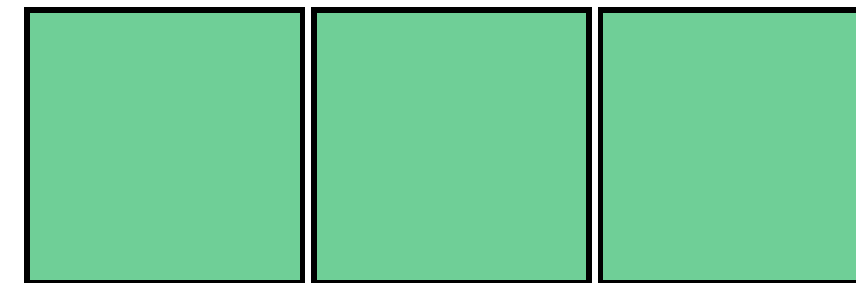


Variáveis Indexadas

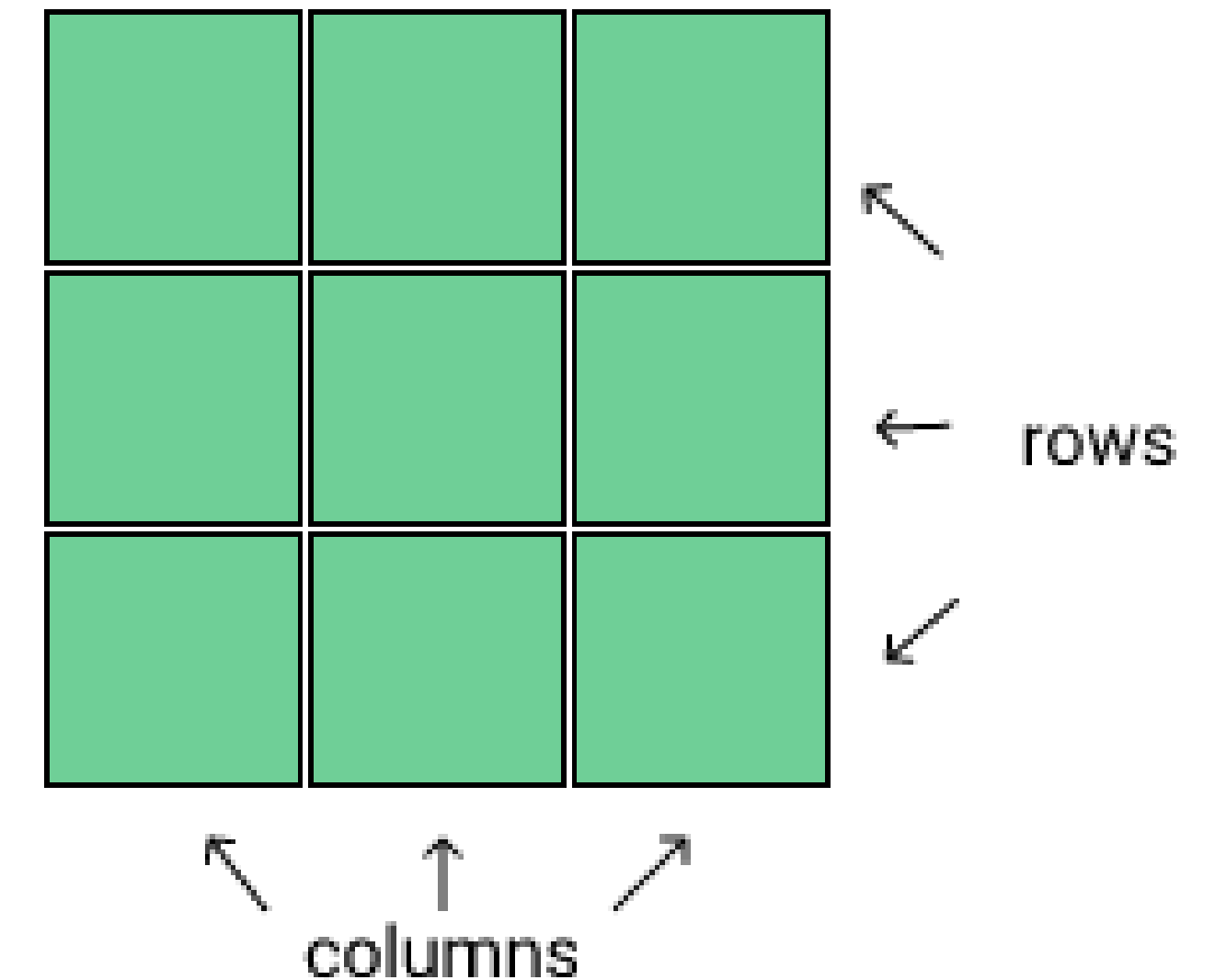
Exemplo:

Como construir um algoritmo que leia os nomes de 500 pessoas e imprima um relatório destes mesmos nomes, mas ordenados alfabeticamente?

Vector



Matrix



Variáveis Indexadas

Solução:

Não seria uma tarefa simples, pois teríamos que definir 500 variáveis do tipo caractere, como é mostrado ao lado.

```
1 Algoritmo "inviavel"  
2 Var  
3 nome1, nome2, ..., nome500: caractere  
4  
5 Inicio  
6 leia(nome1, nome2, ..., nome500)  
7 ....  
8 Fimalgoritmo  
9
```

Variáveis Indexadas

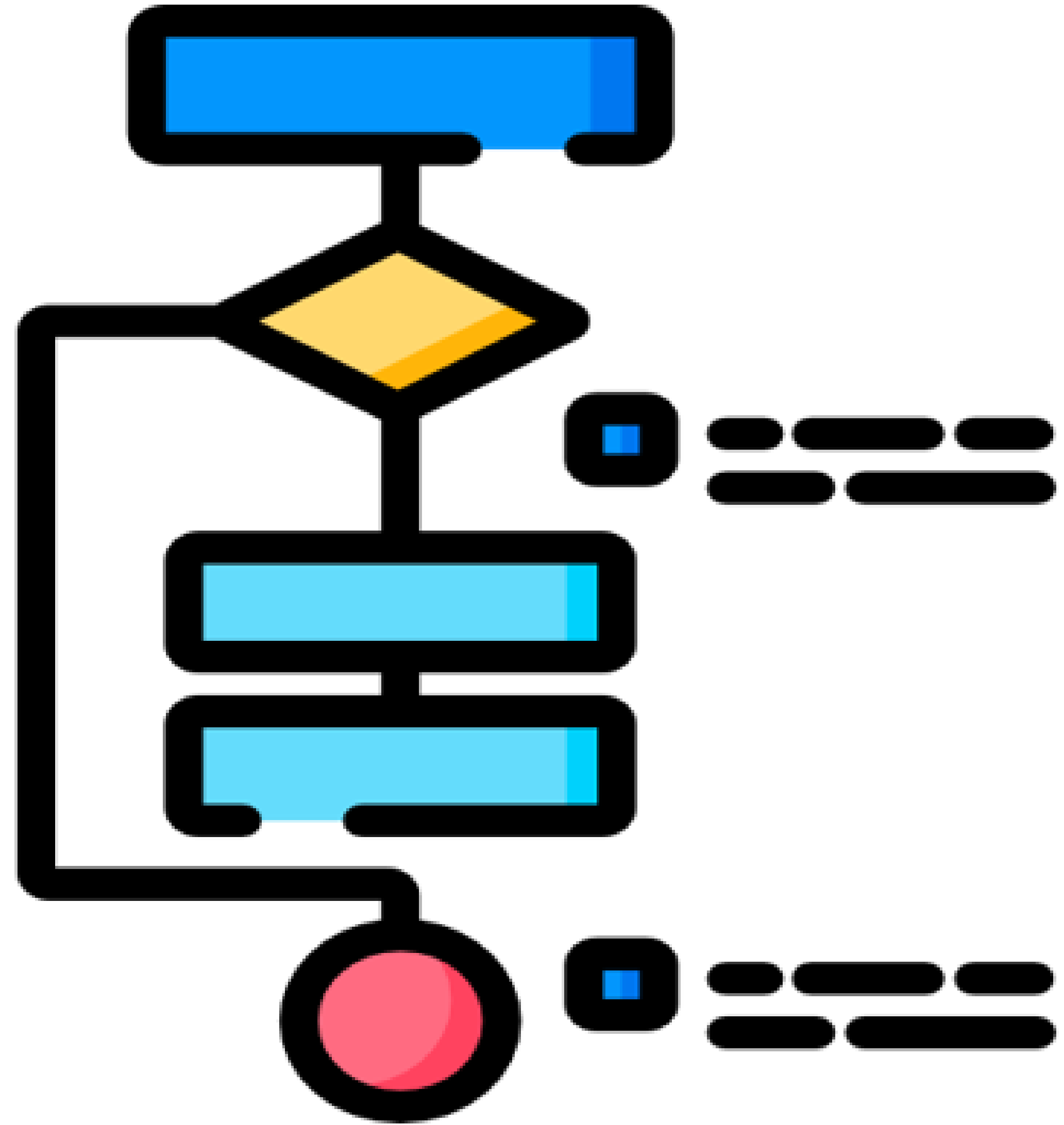
Para resolver problemas como este, e outros, existem as variáveis indexadas. **Vetores e Matrizes.**

A declaração de uma variável indexada corresponde, na verdade, à declaração de várias variáveis cujo identificador difere apenas por um índice.

O índice corresponde a um valor numérico começando por 1.

Vetores

Variáveis Indexadas Unidimensionais



Vetores

Variáveis indexadas com uma **única dimensão**, também conhecidas como **vetores**, são referenciadas por um único índice.

A sintaxe para declaração é:

<Identificador> : vetor [<tamanho>] de < tipo >

tamanho [VI..VF] => Vi = Valor inicial do índice
Vf = Valor final do índice

Vetores

Exemplo:

IDADES: **VETOR** [1..5] **DE** INTEIRO

NOMES: **VETOR** [1..5] **DE** CARACTERE

A declaração acima corresponde à declaração de 10 variáveis:

nomes[1], nomes[2], nomes[3], nomes[4], nomes[5],
idades[1], idades[2], idades[3], idades[4] e idades[5].

Vetores

Atribui-se um valor a um elemento do vetor utilizando o seguinte padrão:

```
< identificador>[Posição] <- valor
```

Exemplos:

```
nomes[1] <- "João da Silva"
```

```
idades[1] <- 35
```

```
nomes[3] <- "Maria Aparecida"
```

```
idades[3] <- idades[1]
```

```
i <- 5
```

```
idades[i] <- 45
```


Vetores

Exemplo 1:

Faça um algoritmo que lê um vetor NUMERO de 6 posições e o escreva. A seguir, ele conta quantos valores de NUMERO são negativos e escreve esta informação.

```
1 Algoritmo "VetorNumeroNegativo"  
2 Var  
3   i, negativo: inteiro  
4   numero: vetor[1..6] de real  
5 Inicio  
6   negativo <- 0  
7   para i de 1 ate 6 faca  
8     leia(numero[i])  
9     se numero[i]<0 entao  
10      negativo <- negativo + 1  
11    fimse  
12  fimpara  
13  escreva("Números negativos: ", negativo)  
14 Fimalgoritmo
```



Vamos
praticar?

Vetores

Exercício 1:

Faça um algoritmo que leia 7 números. Em seguida imprima quantos números são pares e quais são esses números pares. Utilize o conceito de vetor.

102, 104, 106, 108, 110
112, 114, 116, 118, 120
122, 124, 126, 128, 130
132, 134, 136, 138, 140
.
.
.
.
.
.
.
.
.
.
990, 992, 994, 996, 998.

Solução

```
1 algoritmo "TotalPares"
2 var
3   n: vetor[1..7] de Inteiro
4   i, TotPar: Inteiro
5 inicio
6   Para i <- 1 ate 7 faca
7     Escreva("Digite o ", i, "o. valor: ")
8     Leia(n[i])
9   FimPara
10
11   LimpaTela
12
13   Para i <- 1 ate 7 faca
14     Se (n[i] % 2 = 0) entao
15       TotPar <- TotPar + 1
16       EscrevaL("Valor ", n[i], " na posicao ", i, " eh PAR!")
17     FimSe
18   FimPara
19   Escreva("O valor total de números pares eh: ", TotPar)
20 fimalgoritmo
21
```

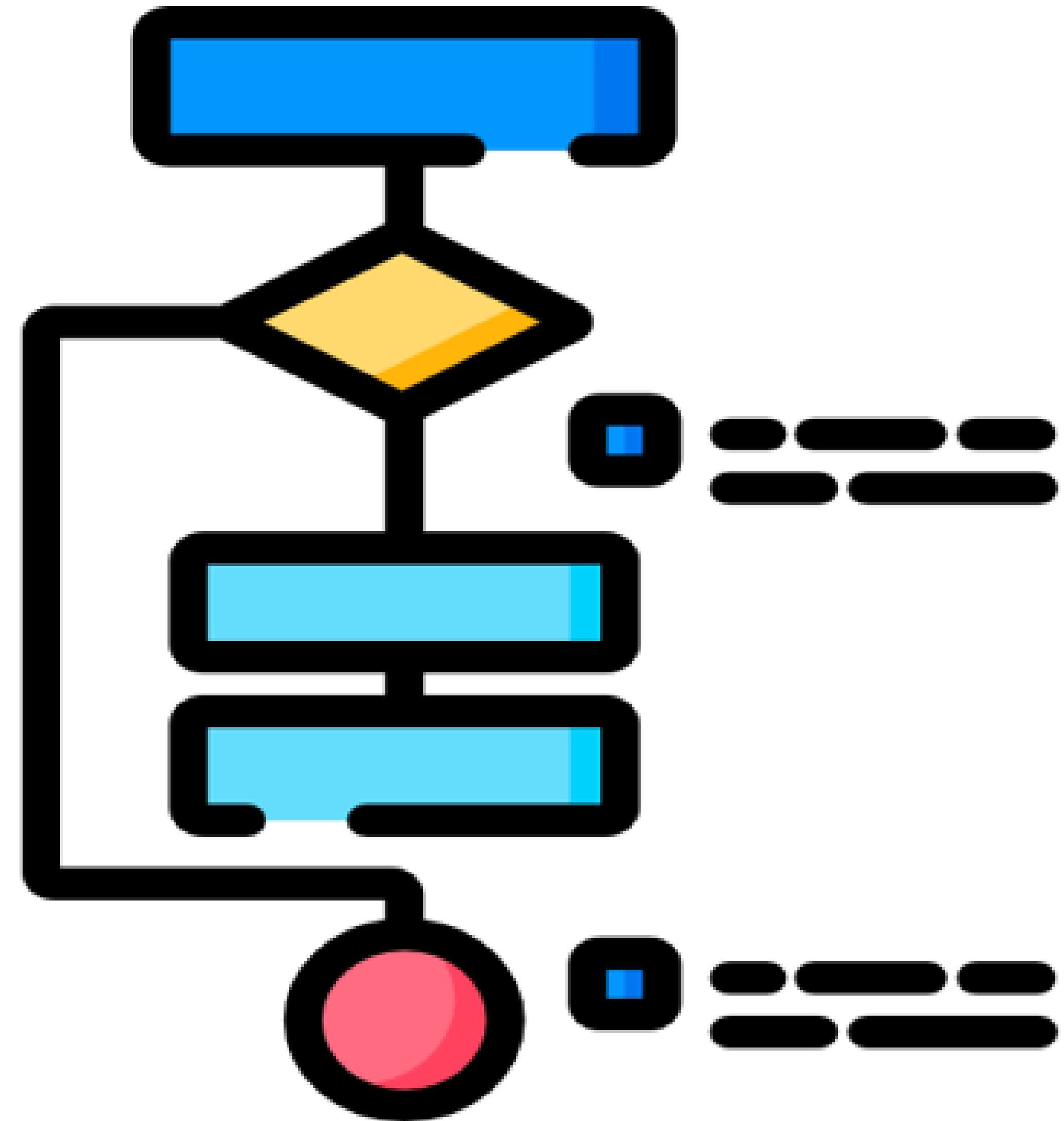



Coffee
time!



Matrizes

Variáveis Indexadas
Bidimensionais



Matrizes

Variáveis indexadas com duas dimensões, também conhecida como matrizes, são referenciadas por dois índices, cada qual começando por 1.

<identificador>: **vetor** [<tamanho1>,<tamanho2>] **de** < tipo >

tamanho [VI..VF]=> Vi= Valor inicial do índice
Vf valor Final do índice.

Matrizes

PESSOAS: **VETOR** [1..2,1..3] **DE** CARACTERE

A declaração acima corresponde à declaração de 6 variáveis:

PESSOAS[1,1], PESSOAS [1,2], PESSOAS[1,3]

PESSOAS[2,1], PESSOAS[2,2], e PESSOAS [2,3].

Matrizes

Para se atribuir um valor a um elemento da matriz devemos utilizar o seguinte padrão:

```
< identificador>[<posição1>,<posição2>] <- <valor>
```

Exemplo:

```
PESSOAS[1,3] <- "Ton"
```

```
PESSOAS[2,1] <- "Isabela"
```

Matrizes

Exemplo 2:

Faça um algoritmo que leia uma matriz(3,3) e calcule a soma de todos os elementos da matriz.

$$\mathbf{Z}_{3 \times 3} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix}$$

Matrizes

```
1 Algoritmo "MatrizSomaTotal"  
2 Var  
3   valores: vetor[1..3,1..3] de real  
4   somaTotal: real  
5   i,j: inteiro  
6 Inicio  
7   Para i de 1 ate 3 faca  
8     Para j de 1 ate 3 faca  
9       Escreval("Digite um valor para a Matriz: ")  
10      Leia(valores[i,j])  
11      somaTotal <- valores[i,j] + somaTotal  
12    fimpara  
13  fimpara  
14  Escreval("A soma total é: ", somaTotal)  
15 Fimalgoritmo
```



Vamos
praticar?

Matrizes

Exercício 2:

Faça um algoritmo que leia uma matriz(3,3) e calcula as somas:

- a) da linha 3;
- b) da coluna 2;
- c) da diagonal principal;
- e) de todos os elementos da matriz.

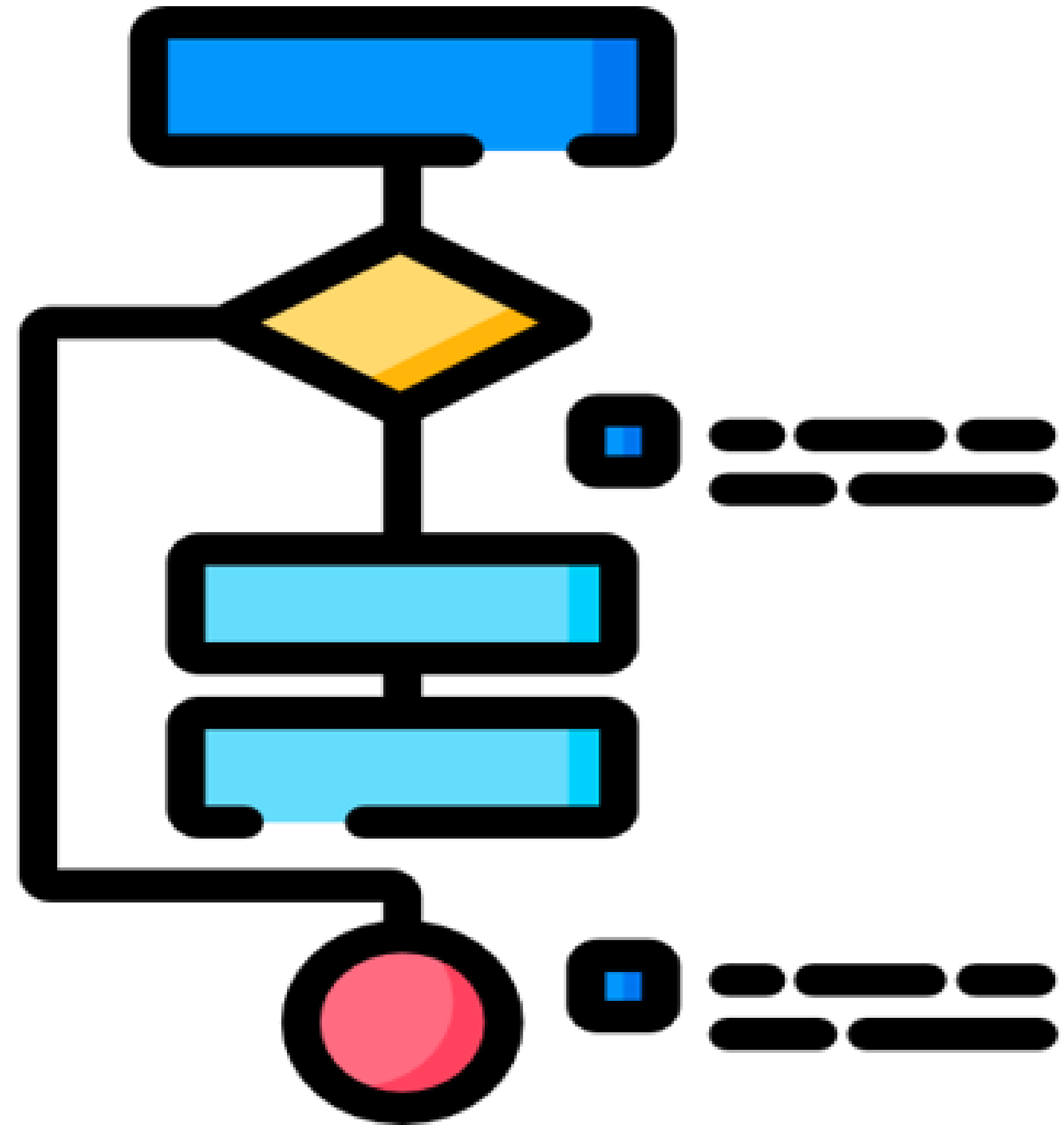
$$\mathbf{Z}_{3 \times 3} = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ Z_{21} & Z_{22} & Z_{23} \\ Z_{31} & Z_{32} & Z_{33} \end{bmatrix}$$

Matrizes

```
1 Algoritmo "MatrizSomas"
2 Var
3   valores: vetor[1..3,1..3] de real
4   somaTotal,somaLinha3,somaColuna2,somaDiag: real
5   i,j: inteiro
6 Inicio
7   somaLinha3 <- 0
8   somaColuna2 <- 0
9   somaDiag <- 0
10  Para i de 1 ate 3 faca
11    Para j de 1 ate 3 faca
12      Escreval("Digite um valor para a Matriz: ")
13      Leia(valores[i,j])
14
15
16
17
18
```

?

Exercícios



Prática

Exercício 3) Faça um Algoritmo para mostrar uma listagem de alunos. Leia os dados (o nome e as duas notas de prova) de 5 alunos. Use o conceito de vetor.

Imprima a média da turma.
Imprima os alunos que tiveram a média acima da turma.



#1168142

Prática

Exercício 4) Faça um algoritmo para gerar uma tabela de jogos de futebol. Leve em consideração que deve existir o jogo de ida e o jogo de volta. Use o conceito de vetor para resolver o exercício.



Prática

Exercício 5) Faça um algoritmo que leia o nome de 10 pessoas e depois mostre uma listagem com nomes que comecem com a letra C.



Prática

Exercícios 6: Faça um algoritmo para corrigir uma prova, a partir de um gabarito. Primeiro leia o gabarito, ou seja, leia a resposta de 5 questões. Em seguida leia o nome de 3 alunos e leia as respectivas respostas das 5 questões marcada por eles. Confira o resultado e imprima na tela, ou seja, a nota alcançada por cada aluno. Imprima também a média da turma.



Prática

Exercícios 7: Faça um algoritmo para preencher uma matriz de quarta ordem e mostre na tela:

- Os elementos da diagonal principal
- Os elementos do triangulo superior
- Os elementos do triangulo inferior
- A matriz completa



Prática

Exercícios 8: Faça um Algoritmo para preencher uma matriz de terceira ordem e mostrar quais são as posições que possuem valores pares.



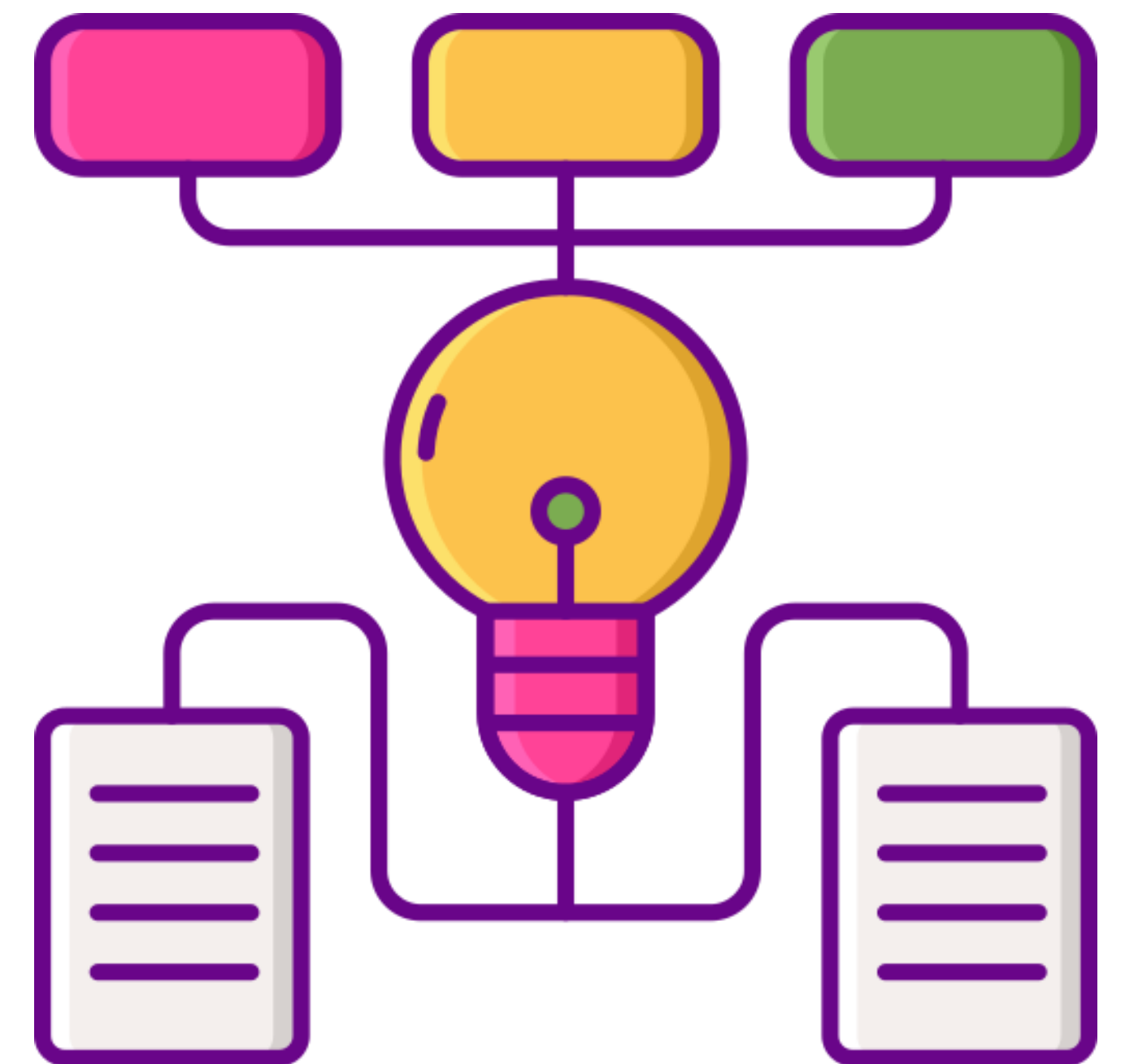
Dica de hoje

Tecclas de Atalho

Abrir (Ctrl-A): Abre um arquivo anteriormente gravado, substituindo o texto presente no editor.

Novo (Ctrl-N): Cria um novo "esqueleto", substituindo o texto presente no editor.

Salvar (Ctrl-S): Grava o texto presente no editor, na primeira vez que um novo texto é gravado, é requisitado um nome e a localização para salvar o arquivo.



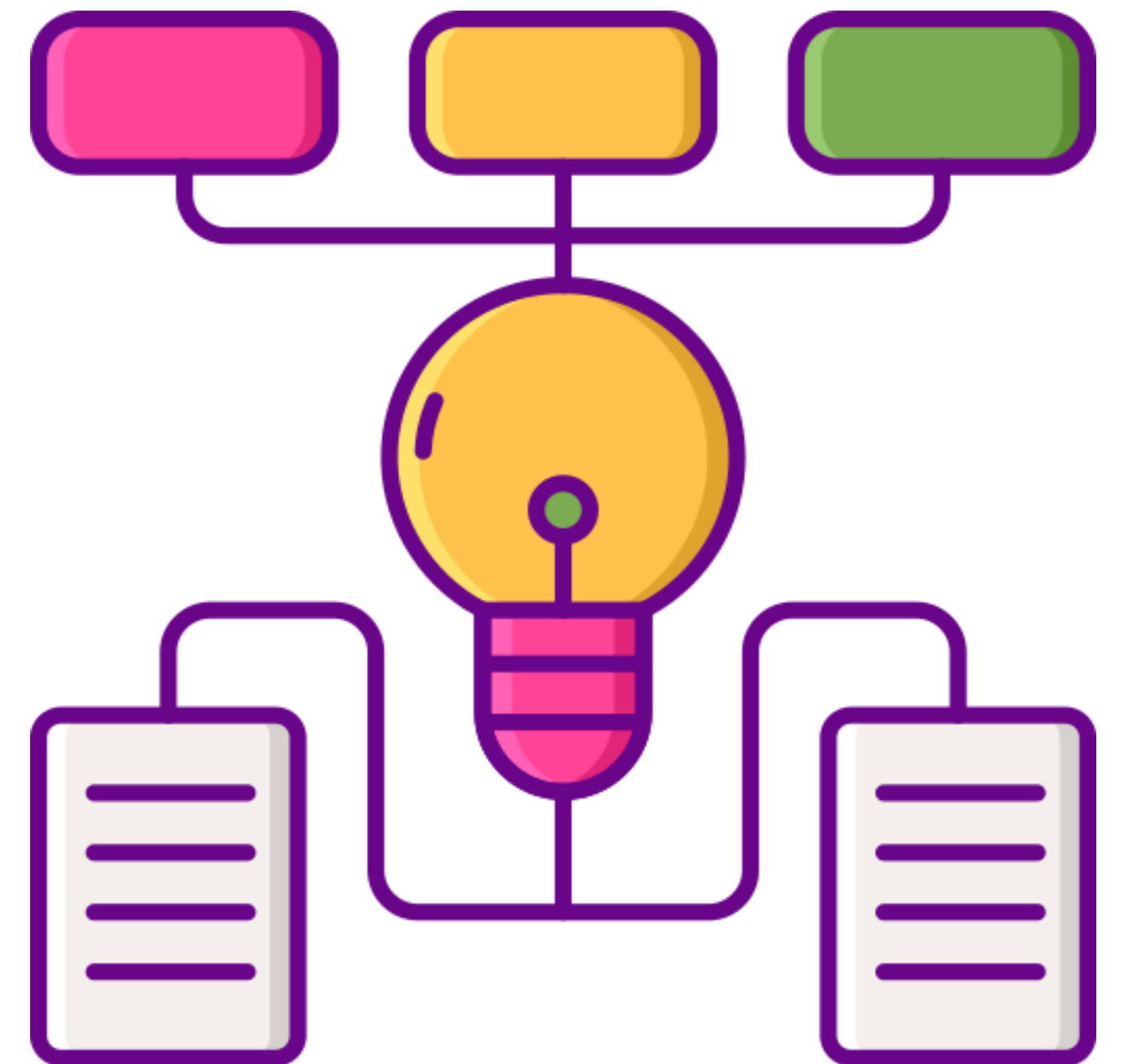
Dica de hoje

Tecclas de Atalho

Desfazer (Ctrl-Z): Desfaz último comando efetuado.

Refazer (Shift-Ctrl-Z): Refaz último comando desfeito.

Localizar (Ctrl-L): Localiza no texto presente no editor determinada palavra especificada.

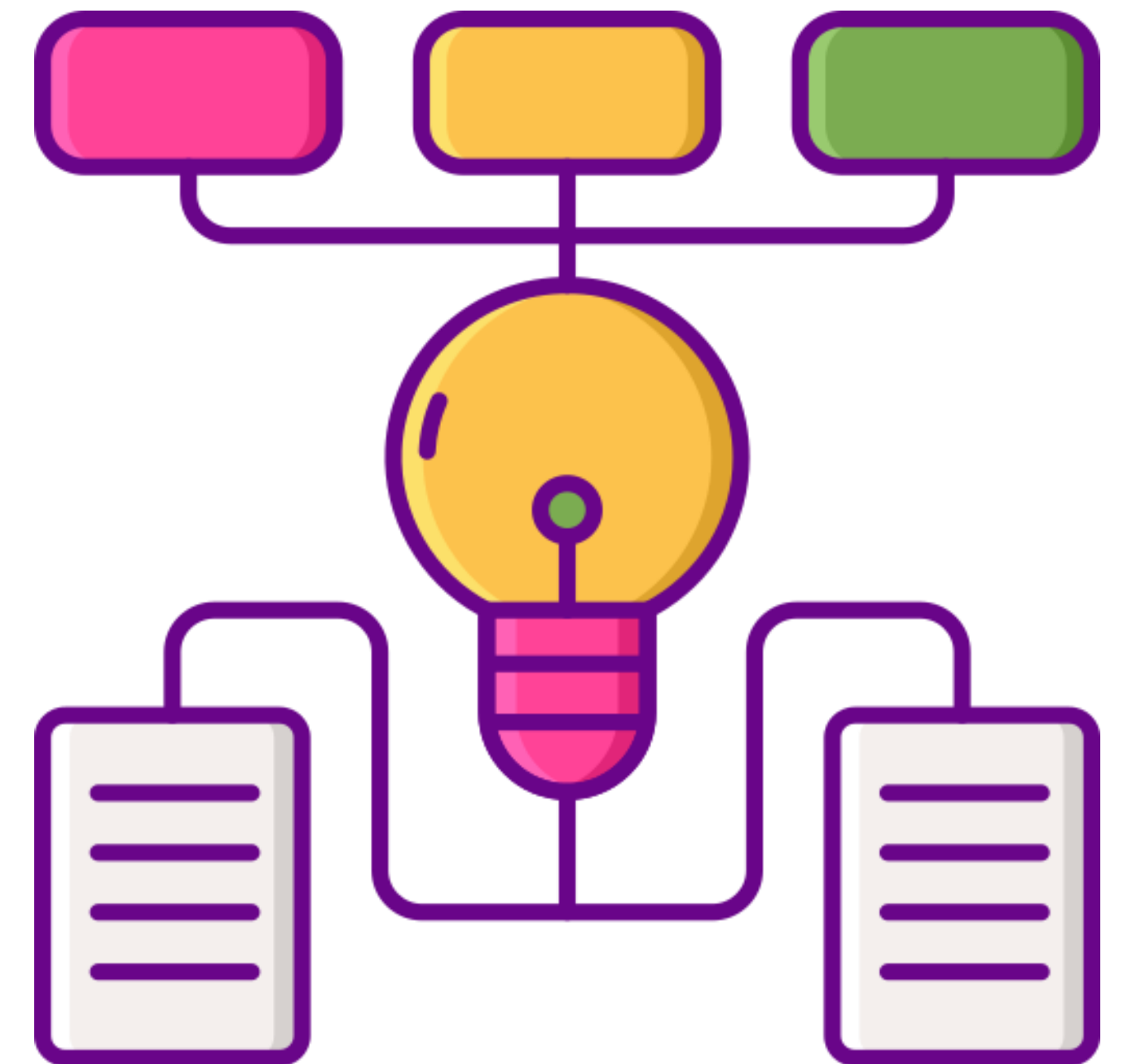


Dica de hoje

Telas de Atalho

Substituir (Ctrl-U): Localiza no texto presente no editor determinada palavra especificada, substituindo-a por outra.

Corrigir Identação(Ctrl-G): Corrige automaticamente a indentação (formatação) do pseudocódigo, tabulando.





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/>

