

UNICARIOCA

ALGORITMOS-II

TEMA-04

TIPOS DE DADOS ESTRUTURADOS VETOR

TIPOS DE DADOS ESTRUTURADOS

A solução de um problema via computador implica em:

- **Identificar itens** (variáveis) **relevantes**.
- **Entender** o **relacionamento** entre os **itens** de dados.

As **formas** nas quais os itens de dados se **relacionam** (logicamente) **definem** diferentes **ESTRUTURAS DE DADOS**.

TIPOS DE DADOS ESTRUTURADOS

ESTRUTURAS DE DADOS podem ser:

- **PRIMITIVAS** \Rightarrow O computador **tem** um conjunto de instruções para **manipulá-las**.
- **NÃO PRIMITIVAS** \Rightarrow O computador **não dispõe** de **instruções** para **manipulá-las** (em geral).

TIPOS DE DADOS ESTRUTURADOS

ESTRUTURAS PRIMITIVAS

Exemplos \Rightarrow

Números reais $\Rightarrow 4+2*3$; $\text{SQRT}(2)$; etc

Números inteiros $\Rightarrow 3+7$; $20/2$;

Strings $\Rightarrow S1$ & $S2$ (concatenar)

etc.

TIPOS DE DADOS ESTRUTURADOS

ESTRUTURAS NÃO PRIMITIVAS

Exemplo-1: **NÚMERO COMPLEXO** \Rightarrow Não é uma estrutura primitiva porque poucas linguagens têm instruções para manipular este tipo de estrutura de dados diretamente.

TIPOS DE DADOS ESTRUTURADOS

ESTRUTURAS NÃO PRIMITIVAS

Exemplo-2: **VETORES**

Var x: vetor [1..10] de inteiro;

y: vetor [1..10] de inteiro;

z: vetor [1..10] de inteiro;

COMO **SOMAR** O **VETOR X** COM O **VETOR Y** E COLOCAR O RESULTADO NO **VETOR Z** ?

PODEMOS FAZER $\Rightarrow z \leftarrow x + y$; ????

Para i de 1 até 10 faça

$z[i] \leftarrow x[i] + y[i]$;

Fim-para;

**PRECISAMOS ENSINAR
A MÁQUINA COMO SOMAR!**

VETOR

MOTIVAÇÃO
var x: real ;

QUANTOS VALORES PODEMOS **ARMAZENAR** NA VARIÁVEL X AO **MESMO TEMPO** ?

x ← 1;
x ← 10; ⇒ **APENAS UM VALOR !**

COMO FAREMOS SE HOVER A NECESSIDADE DE ARMAZENAR **MAIS DE UM VALOR AO MESMO TEMPO** ?

COMO ARMAZENAR **10 VALORES** NA VARIÁVEL X **SIMULTANEAMENTE** ?

VETOR

MOTIVAÇÃO
COMO ARMAZENAR **10 VALORES** NA VARIÁVEL X **SIMULTANEAMENTE** ?
USANDO O CONCEITO DE **VETOR** !

var X: VETOR[**1..10**] de real ;

X	12	26	32	21	14	8	3	19	1	20
POSIÇÃO	1	2	3	4	5	6	7	8	9	10

O QUE ESTÁ **ARMAZENADO** EM X ?
DEPENDE DA **POSIÇÃO** !

PARA IDENTIFICAR A POSIÇÃO PRECISAMOS DE UM **ÍNDICE** !
ASSIM X [**1**] = **12** ; X [**3**] = **32** ; X [**7**] = **3** ; X [**10**] = **20**

VETOR

MOTIVAÇÃO

Suponhamos uma **relação** de **alunos** com **NOME** e **NOTA**.

Assim, **cada entrada** desta lista de dados consiste de **um NOME** e de **uma NOTA**.

PROBLEMA COLOCADO ⇒ **Imprimir o NOME e a NOTA** dos alunos cuja **NOTA** é **maior** que a **MÉDIA da turma**.

VETOR

MOTIVAÇÃO

RELAÇÃO	
NOME	NOTA
BETH	8,0
JOÃO	6,0
ALINE	9,0
ANTÔNIO	5,0
PEDRO	10,0
BIA	4,0

PROBLEMA COLOCADO
⇒ **Imprimir o NOME e a NOTA** dos alunos cuja **NOTA** é **maior** que a **MÉDIA da turma**.

QUAL A ESTRATÉGIA PARA RESOLVER ESSE PROBLEMA ?

VETOR

QUAL A ESTRATÉGIA PARA RESOLVER ESSE PROBLEMA ?

PASSO-1⇒ **DETERMINAR A MÉDIA DAS NOTAS DOS ALUNOS DA TURMA**.

PASSO-2: **COMPARAR MÉDIA DA TURMA COM A NOTA DE CADA ALUNO E IMPRIMIR O NOME E A NOTA DO ALUNO SE SUA NOTA FOR MAIOR DO QUE A MÉDIA DA TURMA**.

PASSO-1 ⇒ DETERMINAR A MÉDIA DAS NOTAS

ALG_01_A
Var Nome: String [30];
Nota, Soma, Média: Real;
Cont: Inteiro;

Início
Cont ← 0;
Soma ← 0;
Leia (Nome, Nota);

Enquanto (Não Fim de Arquivo)
Faça
Cont ← Cont + 1;
Soma ← Soma + Nota;
Leia (Nome, Nota);

Fim-enquanto
Média ← Soma/Cont;
Imprima **Média**;
Fim ALG_01_A

EXEMPLO DA TABELA COM NOME E NOTA			
NOME	NOTA	CONT=0	SOMA=0
BETH	8,0	1	8,0
JOÃO	6,0	2	14,0
ALINE	9,0	3	23,0
ANTÔNIO	5,0	4	28,0
PEDRO	10,0	5	38,0
BIA	4,0	6	42,0
* FIM DE ARQUIVO			

MÉDIA = SOMA/CONT = 42/6
MÉDIA = 7,0

PASSO-2 ⇒ Comparar Média com Nota e imprimir o Nome e a Nota do aluno se sua Nota for maior do que a Média.

ALG_01_B

Const Média ← 7,0;

Var Nome: String [30];

Nota, Média: Real;

Início

Leia(Nome, Nota);

Enquanto (Não Fim de Arquivo)

Faça

Se (Nota > Média)

Então Imprima(Nome,Nota);

Fim-se

Leia (Nome, Nota);

Fim-Enquanto

Fim ALG_01_B

EXEMPLO DA TABELA COM NOME E NOTA

NOME	NOTA	CONT=0	SOMA=0
BETH	8,0	1	8,0
JOÃO	6,0	2	14,0
ALINE	9,0	3	23,0
ANTÔNIO	5,0	4	28,0
PEDRO	10,0	5	38,0
BIA	4,0	6	42,0
*			
* FIM DE ARQUIVO			

O PROBLEMA DO CLIENTE FOI RESOLVIDO ?

O SISTEMA ATENDEU AOS REQUISITOS ?

A SOLUÇÃO É EFICIENTE ?

VOCÊ CONTRATARIA A EMPRESA QUE DESENVOLVEU ESSA APLICAÇÃO ?

SOLUÇÃO ⇒ INEFICIENTE !!!

O CONJUNTO DE DADOS FOI LIDO DUAS VEZES !!!

PENSO, LOGO EXISTO !

DARIA TUDO QUE SEI, PELA

METADE DO QUE IGNORO.

RENÉ DESCARTES

(La Haye en Touraine-1596/Estocolmo-1650)

UNICARIOCA

ALGORITMOS-II

MÓDULO-02

TIPOS DE DADOS

ESTRUTURADOS

VETOR

VÍDEO-02

O QUE DEVEMOS FAZER PARA MELHORAR A SOLUÇÃO?

✓ ASSOCIAR **UMA ÚNICA VARIÁVEL** AO CONJUNTO DE **NOMES**.

✓ ASSOCIAR **UMA ÚNICA VARIÁVEL** AO CONJUNTO DE **NOTAS**.

NOME ⇒ SERÁ UM CONJUNTO ORDENADO DE ELEMENTOS. CADA ELEMENTO (CADA NOME) SERÁ UMA CADEIA DE CARACTERES.

NOTA ⇒ SERÁ UM CONJUNTO ORDENADO DE ELEMENTOS. CADA ELEMENTO (CADA NOTA) SERÁ UM VALOR REAL.

COMO REFERENCIAR UM ITEM ESPECÍFICO?

⇒ USANDO UM **ÍNDICE**.

ASSIM:

NOME[5] ⇒ É O NOME DO **5º ALUNO** NA RELAÇÃO DE NOMES !

NOTA[10] ⇒ É A NOTA DO **10º ALUNO** NA RELAÇÃO DE NOTAS !

VETOR

DEFINIÇÃO ⇒ QUANDO JUNTAMOS UM **ÍNDICE**, AO **NOME** DADO A UM CONJUNTO ORDENADO, A VARIÁVEL RESULTANTE É CHAMADA **VARIÁVEL INDEXADA**.

DEFINIÇÃO ⇒ **VETOR** É UM **CONJUNTO ORDENADO** COM UM **NÚMERO FIXO** DE **ELEMENTOS** DO **MESMO TIPO**.

VETOR - DEFINIÇÃO

var **x**: **vetor** [1..10] de **inteiro**;

⇒ podemos armazenar **10 valores inteiros** na variável **x**

var **nome**: **vetor** [1..10] de **literal** [30];

⇒ podemos armazenar **10 valores literais** (string) na variável **nome** sendo que cada um pode ter até 30 caracteres

var **nota**: vetor [1..10] de **real**;

⇒ podemos armazenar **10 valores reais** na variável **nota**

SOLUÇÃO DO PROBLEMA USANDO VETOR

ALG_04

VarNome: Vetor[1..1000] de String [30] ;

Nota : Vetor[1..1000] de Real;

Média,Soma:Real;

I,N: Inteiro;

Início

Leia (N); { Leitura do Nº de Alunos }

Soma ← 0;

Para I de 1 até N **Faça**

Leia (Nome[I], Nota[I]); { Leitura dos Dados }

Soma ← Soma + Nota[I]; { Cálculo da Soma das Notas }

Fim-Para

Média ← Soma / N; { Cálculo da Média }

Para I de 1 até N **Faça**

Se Nota[I] > Média

Então Imprima (Nome[I], Nota[I]);{ Impressão }

Fim-Se

Fim-Para

Fim ALG_04

EXEMPLO PARA UMA TURMA DE 6 ALUNOS

PARA N = 6 ALUNOS

Para I de 1 até N **Faça**

Leia (Nome[I], Nota[I]);

Soma ← Soma + Nota[I];

Fim-Para

Média ← Soma / N; { Cálculo da Média } (**42/6=7,0**)

Para I de 1 até N **Faça**

Se Nota[I] > **Média**

Então Imprima (Nome[I], Nota[I]) { Impressão }

Fim-Se;

Fim-Para;

I	NOME [I]	NOTA [I]	SOMA=0
1	BETH	8,0	8,0
2	JOÃO	6,0	14,0
3	ALINE	9,0	23,0
4	ANTÔNIO	5,0	28,0
5	PEDRO	10,0	38,0
6	BIA	4,0	42,0

VETOR - OUTROS EXEMPLOS

var **x**: **vetor** [1..10] de **inteiro**;
i: inteiro;

Quantos valores podemos armazenar no vetor x ?

Suponha que desejamos ler os valores abaixo para o vetor x.

9	8	7	10	1	4	3	6	5	2
---	---	---	----	---	---	---	---	---	---

Usamos então o seguinte comando de leitura:

Para i de 1 até 10 faça

leia x[i];

Fim-do-para;

VETOR - OUTROS EXEMPLOS

var **x**: **vetor** [1..10] de **inteiro**;
i: inteiro;

9	8	7	10	1	4	3	6	5	2
---	---	---	----	---	---	---	---	---	---

Após a execução desse comando de leitura o conteúdo do vetor x será:

x	9	8	7	10	1	4	3	6	5	2
Índice (i)	01	02	03	04	05	06	07	08	09	10

x[1] = 9 x[6] = 4
x[2] = 8 x[7] = 3
x[3] = 7 x[8] = 6
x[4] = 10 x[9] = 5
x[5] = 1 x[10] = 2

VETOR - OUTROS EXEMPLOS

var **nome**: **vetor** [1..6] de **literal**[20];
i: inteiro;

Quantos nomes podemos armazenar no vetor **nome** ?

Suponha que desejamos ler os nomes abaixo para o vetor nome.

Beth	João	Maria	Pedro	Antonio	Leda
------	------	-------	-------	---------	------

Usamos então o seguinte comando de leitura:

Para i de 1 até 6 faça

leia nome[i];

Fim-do-para;

VETOR - OUTROS EXEMPLOS

var **nome**: **vetor** [1..6] de **literal**[20];
i: inteiro;

Beth	João	Maria	Pedro	Antonio	Leda
------	------	-------	-------	---------	------

Após a execução desse comando o conteúdo do vetor nome será:

Nome	Beth	João	Maria	Pedro	Antonio	Leda
Índice (i)	01	02	03	04	05	06

nome[1] = Beth
nome[2] = João
nome[3] = Maria
nome[4] = Pedro
nome[5] = Antonio
nome[6] = Leda

Exercício-1: Crie um algoritmo que **leia** 10 valores inteiros para o vetor X, imprima o **maior** elemento e a **posição (índice)** em que ele ocorre.

EXEMPLO - VETOR X

X	9	8	7	10	1	4	3	6	5	2
Índice (i)	01	02	03	04	05	06	07	08	09	10

MAIOR = **10**

POSIÇÃO (ÍNDICE) EM QUE OCORRE = **04**

ALG_M2_EX01

Var X : vetor [1..10] de inteiro; { Declaração de Variáveis}
I, Maior, IND_Maior : Inteiro;

Início

Para I de 1 até 10 **Faça** {Leitura do Vetor}

leia (X[I]);

Fim-para

Maior ← X[1];

IND_Maior ← 1;

Para I de 2 até 10 **Faça**

Se (**Maior** < X[I])

Então Faça

Maior ← X[I];

IND_Maior ← I;

Fim-se;

Fim-para;

Imprima (**Maior** , **IND_Maior**);

Fim ALG_M02_EX01;

X	9	8	7	10	1	4	3	6	5	2
I	01	02	03	04	05	06	07	08	09	10

QUANTAS TROCAS FORAM REALIZADAS ?

ALG_M2_EX01

Var X : vetor [1..10] de inteiro; { Declaração de Variáveis}
I, Maior, IND_Maior : Inteiro;

Início

Para I de 1 até 10 **Faça** {Leitura do Vetor}

leia (X[I]);

Fim-para

Maior ← X[1];

IND_Maior ← 1;

Para I de 2 até 10 **Faça**

Se (**Maior** < X[I])

Então Faça

Maior ← X[I];

IND_Maior ← I;

Fim-se;

Fim-para;

Imprima (**Maior** , **IND_Maior**);

Fim ALG_M02_EX01;

X	9	8	7	10	1	4	3	6	5	2
I	01	02	03	04	05	06	07	08	09	10

I	X[I]	MAIOR	Ind_maior
1	9	9	1
2	8	9	1
3	7	9	1
4	10	10	4
5	1	10	4
6	4	10	4
7	3	10	4
8	6	10	4
9	5	10	4
10	2	10	4

UNICARIOCA

ALGORITMOS-II

MÓDULO-02

TIPOS DE DADOS ESTRUTURADOS

VETOR

VÍDEO-03

Exercício-2: Crie um algoritmo que **leia** 10 valores inteiros para o vetor X, e gere e imprima o vetor Y que contém **1** nas posições **em que X** tem um **valor PAR** e **0** nas posições **em que X** tem um **valor IMPAR**.

EXEMPLO - VETORES X e Y

X	9	8	7	10	6	4	3	1	5	2
Índice [i]	01	02	03	04	05	06	07	08	09	10
Y	0	1	0	1	1	1	0	0	0	1

```

ALG_M2_EX02;
Var  X,Y : vetor [1..10] de inteiro; { Declaração de Variáveis}
      I : Inteiro;
Início
Para I de 1 até 10 Faça      {REGRAS DE NEGÓCIO}
  Leia (X[I]);
  Se (X[I] MOD 2 = 0)
    Então
      Y[I] ← 1
    Senão
      Y[I] ← 0;
  Fim-se;
Fim-para;
Para I de 1 até 10 Faça
  Imprima ( Y[I] );
Fim-para;
Fim ALG_M2_EX02;
  
```

X	9	8	7	10	6	4	3	1	5	2
I	01	02	03	04	05	06	07	08	09	10
Y	0	1	0	1	1	1	0	0	0	1

Exercício-3: Crie um algoritmo que leia um conjunto de números reais (X) e os imprima na ordem inversa da leitura, para os seguintes casos:

- A leitura é controlada por um sentinela (FLAG=9999).
- O primeiro dado a ser lido indica a quantidade de dados propriamente dita.

Valores Lidos ⇒ X

VETOR ⇒ Vet (armazena os valores de X)
QN = quantidade de dados lidos

IMPRIME EM ORDEM INVERSA!
BASTA FAZER O ÍNDICE (I)
VARIAR DE QN (6) ATÉ 1 !

I	6	5	4	3	2	1
Vet[I]	2	8	7	3	6	4

X	QN=0	Vet[QN]
4	1	4
6	2	6
3	3	3
7	4	7
8	5	8
2	6	2
9999		

```

ALG_M02_EX_03A; LÊ IMPRIME
Var QN,I : Inteiro;      {Quantidade de Dados/Índice}
    Vet : Vetor[1..1000] de Real; {Vetor-armazena dados lidos}
    X: Real;              {Valor que será lido}
Início
QN ← 0;                  {Inicializa o índice (QN)}
Leia (X);                {Leitura do primeiro valor}
Enquanto (X <> 9999) Faça {LOOP de Leitura - Flag=9999}
  QN ← QN + 1;           {Incrementa QN}
  Vet[QN] ← X; {Armazena valor lido (X) em Vet na posição QN}
  Leia (X);              {Leitura dos demais números}
Fim-enquanto;
Para I de QN até 1 PASSO -1 Faça {O índice I varia de QN até 1}
  Imprima ( Vet[ I ] );
Fim-para
Fim ALG_M02_EX_03A;
  
```

I	6	5	4	3	2	1
Vet[I]	2	8	7	3	6	4

Exercício-3: Crie um algoritmo que leia um conjunto de números reais e os imprima na ordem inversa da leitura, para os seguintes casos:

- A leitura é controlada por um sentinela (FLAG=9999).
- O primeiro dado a ser lido indica a quantidade de dados propriamente dita (QN).

VETOR ⇒ Vet

QN = 6 (quantidade de dados lidos)

X	4	6	3	7	8	2
Vet[I]	4	6	3	7	8	2
I	01	02	03	04	05	06

X	I	Vet[I]
4	1	4
6	2	6
3	3	3
7	4	7
8	5	8
2	6	2

ALG_M02_EX03B; { Lê Imprime }

Var QN,I: Inteiro
 Vet : Vetor[1..1000] de Real;
 X : Real;

Início

Leia (QN); {Leitura da quantidade de dados - QN}

Para I de 1 até QN Faça

Leia (X);

Vet [I] ← X;

Fim-para;

Para I de QN até 1 PASSO -1 Faça

Imprima (Vet [I]);

Fim-para;

Fim ALG_M02_EX03B;

Vet[I]	4	6	3	7	8	2
I	01	02	03	04	05	06

Exercício-3: Crie um algoritmo que leia um conjunto de números reais e os imprima na ordem inversa a da leitura, para os seguintes casos:

a) A leitura é controlada por um sentinela (FLAG=9999).

b) O primeiro dado a ser lido indica a quantidade de dados propriamente dita.

Como seria a solução SEM USAR A VARIÁVEL X para ler os dados ? Podemos NÃO USAR A VARIÁVEL X e ler os dados diretamente para o vetor Vet ?

VETOR ⇒ Vet

QN = quantidade de dados lidos

DADOS	4	6	3	7	8	2
-------	---	---	---	---	---	---

DADOS	QN=0	Vet[QN]
4	1	4
6	2	6
3	3	3
7	4	7
8	5	8
2	6	2
9999		

ALG_M02_EX_03C; LÊ IMPRIME

Var QN,I : Inteiro; {Quantidade de Dados/Índice}
 Vet : Vetor[1..1000] de Real; {Vetor-armazena dados lidos}

Início

QN ← 1; {Inicializa o índice (QN)}

Leia (Vet[1]); {Leitura do primeiro valor}

Enquanto (Vet[QN]<>9999) Faça {LOOP de Leitura - Flag=9999}

QN ← QN + 1;

{Incrementa QN}

Leia (Vet[QN]); {Armazena valor lido em Vet na posição QN}

Fim-enquanto;

QN ← QN - 1; {Acerto na quantidade de dados}

Para I de QN até 1 PASSO -1 Faça {O índice I varia de QN até 1}

Imprima (Vet [I]);

Fim-para

Fim ALG_M02_EX_03C;

I	6	5	4	3	2	1
Vet[I]	2	8	7	3	6	4

ALG_M02_EX_03C; LÊ IMPRIME

Var QN,I : Inteiro;
 Vet : Vetor[1..1000] de Real;

EXEMPLO COM 1 DADO LIDO

Início

QN ← 1;

Leia (Vet[1]);

Enquanto (Vet[QN]<>9999) Faça

QN ← QN + 1;

Leia (Vet[QN]);

QN ← QN - 1;

Fim-enquanto;

EXEMPLO COM 0 DADO LIDO

QN ← QN - 1;

Para I de QN até 1 PASSO -1 Faça

Imprima (Vet [I]);

Fim-para

Fim ALG_M02_EX_03C;

DADOS	QN=1	Vet[QN]
4	1	4
9999	2	9999

DADOS	QN=1	Vet[QN]
9999	1	9999

QN ← QN - 1;

ALG_M02_EX03D; { Lê Imprime }

Var QN,I: Inteiro
 Vet : Vetor[1..1000] de Real;
 Início

Leia (QN); {Leitura da quantidade de dados - QN}

Para I de 1 até QN Faça

Leia (Vet [I]);

Fim-para;

Para I de QN até 1 PASSO -1 Faça

Imprima (Vet [I]);

Fim-para;

Fim ALG_M02_EX03D;

Vet [I]	4	6	3	7	8	2
I	01	02	03	04	05	06

Não há nada no mundo que esteja **melhor** repartido do que a **razão**: **todas** as **pessoas** estão **convencidas** de que a **tem** de **sobra**.

RENÉ DESCARTES

(La Haye en Touraine-1596/Estocolmo-1650)