





# Unidade IV: Vetores

Prof. Me. Pietro Martins de Oliveira

# Introdução

- Estruturas de dados homogêneas
  - São estruturas de dados que agrupam diversas informações do MESMO tipo em uma única variável
    - **Variável simples:**
      - Apenas um dado é acessível
    - **Agregados homogêneos:**
      - Vários dados são acessíveis
  - Podem ser unidimensionais (vetor) ou multidimensionais (matrizes)

## Introdução

- É usual manipular conjuntos de valores, em detrimento de dados isolados;
  - Exemplos: notas, preços, produtos, nomes, etc
- Para manipular um conjunto de dados do mesmo tipo há uma estrutura denominada **vetor**.
  - Analogia:
    - Um vetor é como um trem - cada posição do vetor é um vagão

# Vetores

- Também conhecido como:
  - Arranjo
  - *Array*
  - Agregado homogêneo unidimensional
- “São matrizes de uma única linha”
- Vetores em pseudocódigo:
  - São declarados de maneira similar a uma variável comum, porém, adiciona-se o tamanho
  - Têm um tamanho máximo **N**
  - O primeiro elemento encontra-se na posição **1**
  - O último elemento encontra-se na posição **N**

# Vetores

- Também conhecido como:
  - Arranjo
  - *Array*
  - Agregado homogêneo unidimensional
- “São matrizes de uma única linha”
- Vetores em pseudocódigo:
  - São declarados de maneira similar a uma variável comum, porém, adiciona-se o tamanho
  - Têm um tamanho máximo **N**
  - O primeiro elemento encontra-se na posição **1**
  - O último elemento encontra-se na posição **N**

# Vetores

- Sintaxe de declaração de um vetor em pseudocódigo*

`<nome_variavel>: Vetor[<dimensão>] de <tipo>`

- Exemplo de criação de um vetor para armazenar até 4 (quatro) valores do tipo real:

`notas: Vetor[1..4] de real`

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Vetores

- **Acesso aos elementos do vetor: índices**
- *Sintaxe em pseudocódigo:*

`<nome_do_vetor>[<índice>]`

- Exemplos

1) `notas[1] ← 10.0`

2) `escreva(notas[1])`

3) `soma ← soma + notas[i]`



## Exemplo

**Algoritmo** meu\_vetor1

**Início**

**Fim**

## Exemplo

**Algoritmo** meu\_vetor1

**Var** notas: **Vetor**[1..4] de **real**

**Início**

**Fim**

## Exemplo

**Algoritmo** meu\_vetor1

**Var** notas: **Vetor**[1..4] de **real**

**Início**

    escreva("Digite a nota 1:")

    leia(notas[1])

**Fim**

## Exemplo

**Algoritmo** meu\_vetor1

**Var** notas: **Vetor**[1..4] de **real**

**Início**

escreva("Digite a nota 1:")

leia(notas[1])

escreva("Digite a nota 2:")

leia(notas[2])

**Fim**

## Exemplo

**Algoritmo** meu\_vetor1

**Var** notas: **Vetor**[1..4] de **real**

**Início**

escreva("Digite a nota 1:")

leia(notas[1])

escreva("Digite a nota 2:")

leia(notas[2])

escreva("Digite a nota 3:")

leia(notas[3])

**Fim**

## Exemplo

**Algoritmo** meu\_vetor1

**Var** notas: **Vetor**[1..4] de **real**

**Início**

escreva("Digite a nota 1:")

leia(notas[1])

escreva("Digite a nota 2:")

leia(notas[2])

escreva("Digite a nota 3:")

leia(notas[3])

escreva("Digite a nota 4:")

leia(notas[4])

**Fim**

## Exemplo

**Algoritmo** meu\_vetor2

**Var** notas: **Vetor**[1..4] de **real**

**Inicio**

**Para** i de 1 até 4 **passo** 1 **faça**

        escreva("Digite a nota ", i, ":")

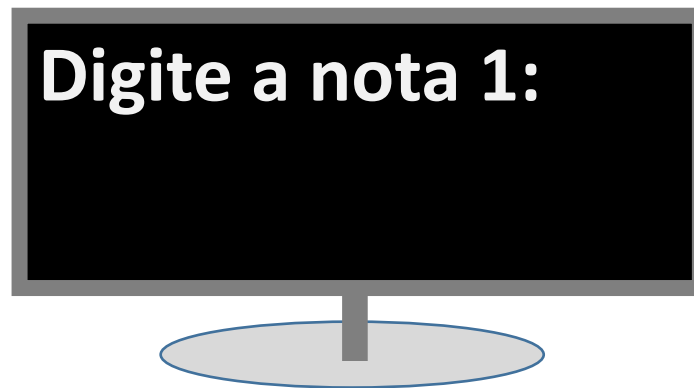
        leia(notas[i])

**Fim\_para**

**Fim**

## Teste de mesa

O que seria exibido para o usuário:



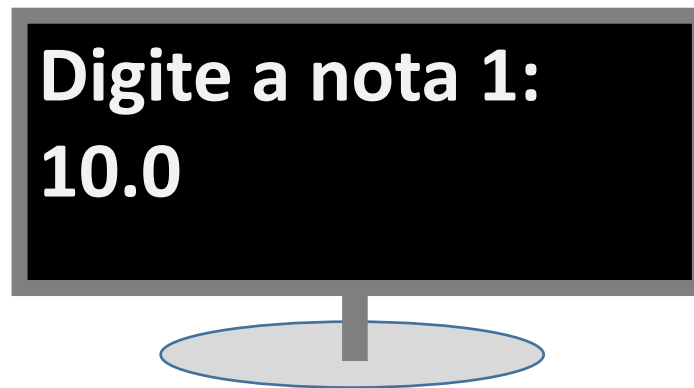
O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo				



## Teste de mesa

O que seria exibido para o usuário:

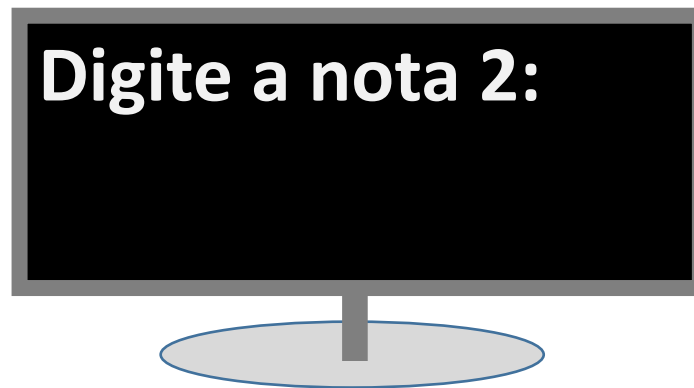


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0			

## Teste de mesa

O que seria exibido para o usuário:

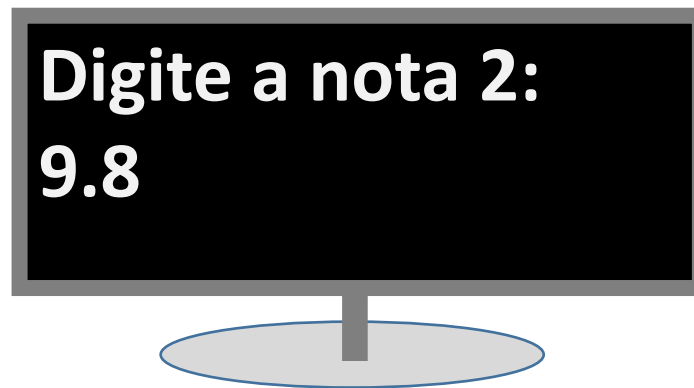


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0			

## Teste de mesa

O que seria exibido para o usuário:

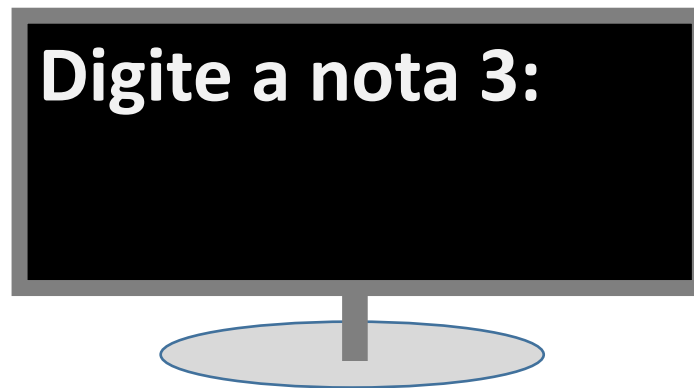


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0	9.8		

## Teste de mesa

O que seria exibido para o usuário:

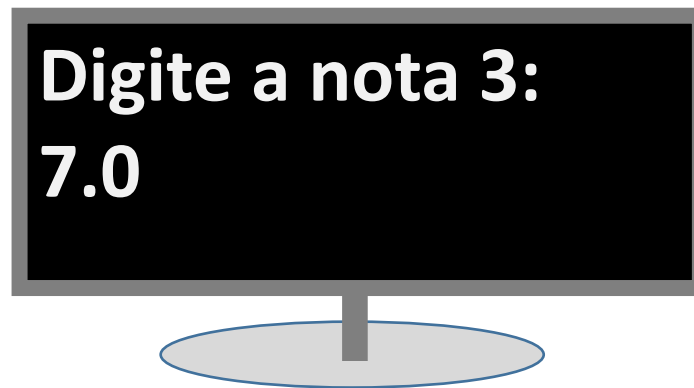


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0	9.8		

## Teste de mesa

O que seria exibido para o usuário:

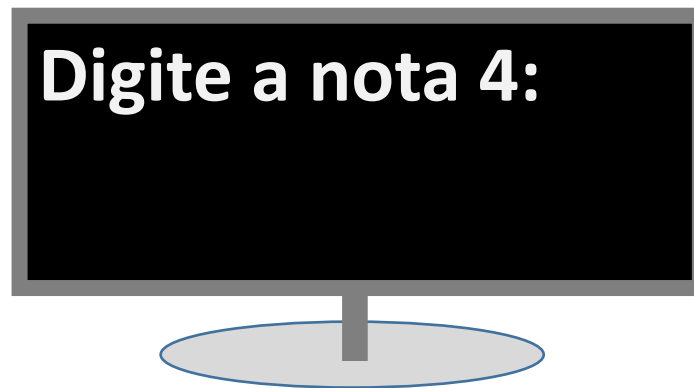


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0	9.8	7.0	

## Teste de mesa

O que seria exibido para o usuário:

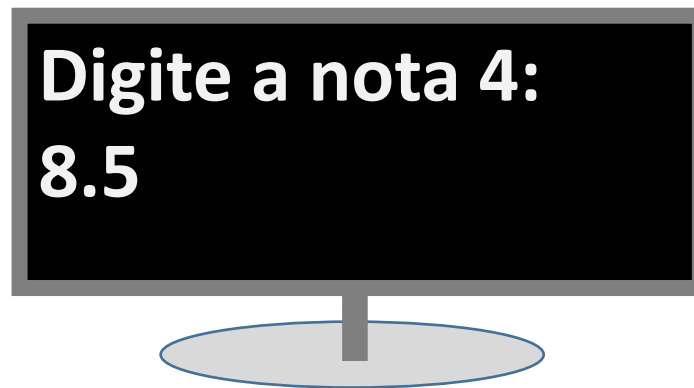


O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0	9.8	7.0	

## Teste de mesa

O que seria exibido para o usuário:



O que aconteceria na memória do computador:

Índice	1	2	3	4
Conteúdo	10.0	9.8	7.0	8.5

*Fim da execução!*

## Busca em vetores

- Técnicas para recuperar informações mantidas dentro de um vetor de dados
- Um dos principais algoritmos é a Busca Sequencial
  - Simples implementação
  - Percorre todo o vetor a partir do início
  - Interrompe quando encontrar o dado desejado
  - Interrompe caso atinja o final do vetor
- Existem diversos outros algoritmos de busca
  - Busca indexada, busca binária, etc



## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice		2	3	4
Conteúdo	9,8	7,0	8,5	

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	7,0	7,0	8,5



## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5

## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5



## Busca em vetores

Buscando pelo valor 7.0 no vetor a seguir:

**Valor 7,0 encontrado na posição 3 do vetor**

Índice	1	2	3	4
Conteúdo	10,0	9,8	7,0	8,5



# Algoritmos de Lógica de Programação I

ITL 8002

ABSTRACTION  
PRESENTATION  
MANUALS

11 14  
2:PRINT"XXXXXXXXXX";

END

(32) THEN PRINT MB\$

PRINT"XXXXXXXXXX";  
STEP 2

PRINT"XXXXXXXXXX" THEN PRINT"E"

PRINT"XXXXXXXXXX";  
STEP 2

PRINT"XXXXXXXXXX" THEN PRINT"E"

END





# Unidade IV : Matrizes

Prof. Me. Pietro Martins de Oliveira

## Introdução

- **Até aqui aprendemos a manipular apenas *arrays* simples:**
  - Estrutura de dados unidimensional;
  - Controlada através de um único índice
- **Consegue imaginar as situações em que é necessário mais de um índice?**
  - Tabelas, gráficos 3D, aplicações contábeis, processamento de imagens e visão computacional, jogos, aplicações científicas, projetos de engenharia etc.

## Introdução

- Considere a tabela:

<div><div>ano</div><div>produto</div></div>	2010	2011	2012
<u>Arroz</u>	6,00	6,50	7,00
<u>Feijão</u>	4,00	4,35	5,25
<u>Macarrão</u>	3,50	3,30	3,75

- Aplicações com tabelas são inerentemente matriciais
  - Solução: MATRIZES

# Matrizes

- **Definição:** matrizes são estruturas de dados multidimensionais que necessitam de mais de um índice para serem manipuladas.
- São parecidas com vetores
  - Semelhanças: os elementos de uma matriz são referenciados por um mesmo nome
  - Diferença: possuem dois ou mais índices
  - **Bidimensionais:** um índice para linhas e outro índice para colunas
  - Multidimensionais: podem possuir 2, 3, ... K índices



# Matrizes

- Definição matemática: uma **matriz** é um **arranjo tabular** de  $M \times N$  valores, onde  $M$  é o número de **linhas** e  $N$  é o número de **colunas**.
- Os elementos de uma matriz são acessados por dois índices:
  - $i$  – geralmente associado às **linhas**
  - $j$  – geralmente associado às **colunas**



# Matrizes

## Sintaxe de declaração de uma matriz

`<nome_m>: Vetor[<d1>, <d2>, ..., <dN>] de <tipo>`

Exemplo de criação de uma matriz bidimensional:

`Mat: Vetor[1..3, 1..3] de inteiro`

$$\text{Mat} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

# Matrizes

## Sintaxe de acesso a um elemento da matriz

`<nome_da_matriz>[<ind1><ind2>, ..., <indN>]`

## Exemplo de acesso a elementos da matriz

- `Mat[1,1]`
- `Mat[3,3]`
- `Mat[1,3]`
- `Mat[2,1]`

1	2	3
4	5	6
7	8	9

## Exemplo

Desenvolver um algoritmo que realize a soma dos elementos da seguinte matriz. Considere que a matriz já foi preenchida com valores, ou seja, já está armazenada na memória, da seguinte forma:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**

**Inicio**

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Para** i **de** 1 **até** 3 **passo** 1 **faça**

**Fim\_para**

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Para** i de 1 até 3 passo 1 **faça**

**Para** j de 1 até 3 passo 1 **faça**

**Fim\_para**

**Fim\_para**

**Fim**



## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Para** i de 1 até 3 passo 1 **faça**

**Para** j de 1 até 3 passo 1 **faça**

soma  $\leftarrow$  soma +

**Fim\_para**

**Fim\_para**

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Para** i de 1 até 3 passo 1 **faça**

**Para** j de 1 até 3 passo 1 **faça**

soma  $\leftarrow$  soma + mat[i,j]

**Fim\_para**

**Fim\_para**

**Fim**

## Exemplo

**Algoritmo** somaElementos

**Var** mat: Vetor[1..3, 1..3] de **inteiro**  
i, j, soma: **inteiro**

**Inicio**

soma  $\leftarrow$  0

**Para** i de 1 até 3 passo 1 **faça**

**Para** j de 1 até 3 passo 1 **faça**

soma  $\leftarrow$  soma + mat[i,j]

**Fim\_para**

**Fim\_para**

escreva("A soma é ", soma)

**Fim**



# Algoritmos de Lógica de Programação I





# Unidade IV: Registros

Prof. Me. Pietro Martins de Oliveira

## Introdução

- Os tipos primitivos são eficientes em todas as situações?
  - Será que não existe algum caso em que é preciso criar um novo tipo?
- Imagine uma aplicação de controle de estoque de produtos:
  - É preciso armazenar várias informações
  - Código, nome do produto, quantidade em estoque, valor de compra, valor de venda, lucro, observações sobre o produto

# Introdução

- **Estruturas de Dados Heterogêneas**
  - Também conhecidas como **registros**, tipos de dados compostos, *structs* (C).
- **Exemplo: Controle de estoque (Produto)**
  - Código: número inteiro
  - Nome do produto: texto
  - Quantidade estocada: número real
  - Valor da compra: número real
  - Valor de venda: número real
  - Lucro: número real
  - Observações sobre o produto: texto



## Registro

- *Sintaxe de definição em pseudocódigo:*

**Tipo**

`<identificador> = registro`

`<campos e seus tipos>`

**Fim\_registro**

- Identificador: nome do novo tipo
- Campos: relação dos campos que compõem o novo tipo

## Exemplo

- Considerando uma aplicação de controle de estoque, crie um algoritmo para que seja possível armazenar as seguintes informações sobre um produto:
  - Código do produto
  - Nome do produto
  - Quantidade do produto em estoque
  - Valor unitário do produto

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

**Fim\_registro**

**Var**

**Início ...**

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

**Fim\_registro**

**Var**

**Início ...**

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

**Fim\_registro**

**Var**

**Início ...**

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

**Fim\_registro**

**Var**

**Início ...**

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

quantidade: **inteiro**

**Fim\_registro**

**Var**

**Início ...**

## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

quantidade: **inteiro**

valor: **real**

**Fim\_registro**

**Var**

**Início ...**



## Exemplo – Declaração de um registro, seus campos e uma variável heterogênea

**Algoritmo** exemploRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

quantidade: **inteiro**

valor: **real**

**Fim\_registro**

**Var**

p1: produto

**Início ...**

## Exemplo – Acessando os campos de um registro

. . .

**Var**

p1: produto

**Início**

**Fim**

## Exemplo – Acessando os campos de um registro

...

**Var**

p1: produto

**Início**

Escreva("Digite o código:")

**Fim**

## Exemplo – Acessando os campos de um registro

...

**Var**

p1: produto

**Início**

Escreva("Digite o código:")

Leia(p1.codigo)

**Fim**

## Exemplo – Acessando os campos de um registro

. . .

**Var**

p1: produto

**Início**

Escreva("Digite o código:")

Leia(p1.codigo)

Escreva("Insira o valor:")

**Fim**

## Exemplo – Acessando os campos de um registro

...

**Var**

p1: produto

**Início**

Escreva("Digite o código:")

Leia(p1.codigo)

Escreva("Insira o valor:")

Leia(p1.valor)

**Fim**

## Registros & Vetores

- O exemplo anterior trata apenas um produto.
  - *Como criar uma lista de produtos?*
- **Registros** podem ser **combinados** com **vetores**:
  - i. Declara-se um novo registro
  - ii. Cria-se um vetor de registros (assim como se cria uma variável do tipo registro)

## Exemplo – Vetor de registros

**Algoritmo** vetorDeRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

quantidade: **inteiro**

valor: **real**

**Fim\_registro**

**Var**



## Exemplo – Vetor de registros

**Algoritmo** vetorDeRegistros

**Tipo**

produto = **registro**

codigo: **inteiro**

nome: Vetor[1..50] de **caractere**

quantidade: **inteiro**

valor: **real**

**Fim\_registro**

**Var** lista\_prod: **Vetor**[1..10] de produto

## Exemplo – Vetor de registros

**Início**

```
escreva("Digite o código produto 1:")  
leia(lista_prod[1].codigo)
```

...

**Fim**

## Exemplo – Vetor de registros

### Início

```
escreva("Digite o código produto 1:")  
leia(lista_prod[1].codigo)  
escreva("Insira o valor produto 1:")  
leia(lista_prod[1].valor)
```

...

### Fim

## Exemplo – Vetor de registros

### Início

```
escreva("Digite o código produto 1:")  
leia(lista_prod[1].codigo)  
escreva("Insira o valor produto 1:")  
leia(lista_prod[1].valor)  
escreva("Digite o código produto 2:")  
leia(lista_prod[2].codigo)
```

...

### Fim

## Exemplo – Vetor de registros

### Início

```
escreva("Digite o código produto 1:")
leia(lista_prod[1].codigo)
escreva("Insira o valor produto 1:")
leia(lista_prod[1].valor)
escreva("Digite o código produto 2:")
leia(lista_prod[2].codigo)
escreva("Insira o valor produto 2:")
leia(lista_prod[2].valor)
...
```

### Fim

