

Estrutura de Dados II



Unidade 1 - Revisão

Prof. Sandro T. Pinto



Uma estrutura de dados pode ser definida como sendo **uma coleção de variáveis**, podendo ser **tipos iguais ou diferentes**, reunidas sob **um único nome**.

Vários autores denominam estrutura de dados como sendo **registros**

Mizrahi 2006



Unidade 1

Estrutura de Repetição

Estrutura Enquanto (While):

A sintaxe do comando Enquanto no algoritmo é a seguinte:

- Enquanto uma **expressão** for verdadeira, faça o **bloco** e FimEnquanto.

```
1 public class Tableless {  
2  
3     public static void main(String[] args) {  
4  
5         int x = 15;  
6  
7         while (x < 18) {  
8             System.out.println("Você não tem permissão para entrar");  
9             x++;  
10        }  
11  
12    }  
13 }
```

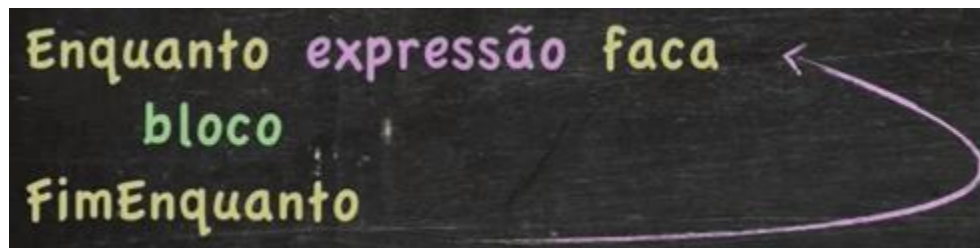
Unidade 1

Estrutura de Repetição

Estrutura Enquanto (While):

Ao entrar no Enquanto, a cada vez ele vai testar uma expressão

- Se a expressão for verdadeira, será executado um bloco, que pode ter quantos comandos forem necessários.
- E ao atingir o FimEnquanto, ele irá voltar ao comando enquanto, esse processo é chamado de looping.



Unidade 1

Estrutura de Repetição

Estrutura **Enquanto**:

- Exemplo contar até 10.

```
algoritmo "contador"  
  
var  
    n: inteiro  
    inicio  
    n <- 0  
    Enquanto (n <= 10) faça  
        Escreval(n)|  
        n <- n + 1  
    FimEnquanto  
  
fimalgoritmo
```

Unidade 1

Estrutura de Repetição

Estrutura Enquanto:

- Exercício contar de 10 a 0: algoritmo "contagem_regressiva"

var

n: inteiro

inicio

n <- 10

enquanto (n >= 0) faça

escreval (n)

n <- n - 1

fimenquanto

fimalgoritmo

Unidade 1

Estrutura de Repetição

Estrutura Enquanto:

Exercício contar de 0 a 100 pulando de 10 em 10:

```
algoritmo "contador"  
  
var  
    n: inteiro  
    inicio  
    n <- 0  
    Enquanto (n <= 100) faca  
        Escreval(n)  
        n <- n + 10  
    FimEnquanto  
  
fimalgoritmo
```


Unidade 1

Estrutura de Repetição

Estrutura Enquanto:

Exercício contar de 0 a 10 e mostrar somente os números pares:

```
algoritmo "contador"
```

```
var
```

```
  n: inteiro
```

```
  inicio
```

```
  n <- 0
```

```
  Enquanto (n <= 10) faca
```

```
    Escreval(n)
```

```
    n <- n + 2
```

```
  FimEnquanto
```

```
fimalgoritmo
```

Unidade 1

Estrutura de Repetição

Estrutura **Enquanto**:

Exercício contar o valor até o número definido usuário.

```
algoritmo "contador"

var
    n, cont: inteiro
    inicio
    Escreva ("Até quanto quer contar:")
    leia(cont)
    n <- 0
    Enquanto (n <= cont) faca
        Escreval(n)
        n <- n + 1
    FimEnquanto

fimalgoritmo
```

Unidade 1

Estrutura de Repetição

Estrutura **Para (for)**:

Para é uma estrutura de repetição com variável de controle, o que permite que você execute o bloco interno uma quantidade determinada de vezes.

```
1 public class Tableless {  
2  
3     public static void main(String[] args) {  
4  
5         for(int x = 0; x<1000; x++){  
6             System.out.println("Valor: " + x);  
7         }  
8     }  
9 }
```

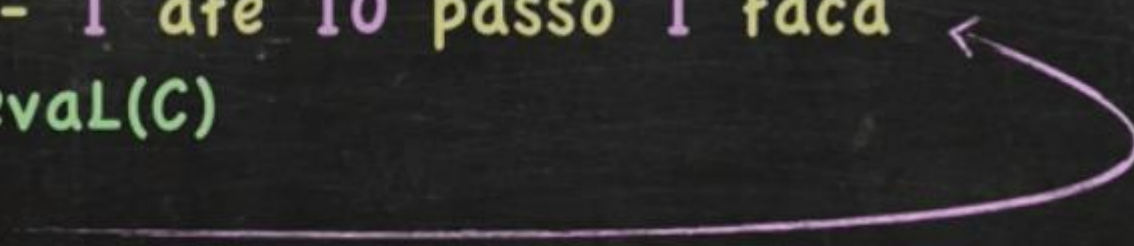
Unidade 1

Estrutura de Repetição

Estrutura Para (for):

```
Para variavel <- inicio ate fim [passo salto] faca  
    Bloco  
FimPara
```

```
Para C <- 1 ate 10 passo 1 faca  
    EscrevaL(C)  
FimPara
```



Unidade 1

Estrutura de Repetição

Estrutura Para (for):

Exemplos:

```
algoritmo "contnumeros"
```

```
var
```

```
    n: inteiro
```

```
inicio
```

```
    Para n <- 1 ate 10 passo 2 faca
```

```
        escreval(n)
```

```
    Fimpara
```

```
fimalgoritmo
```

```
algoritmo "contnumeros"
```

```
var
```

```
    n: inteiro
```

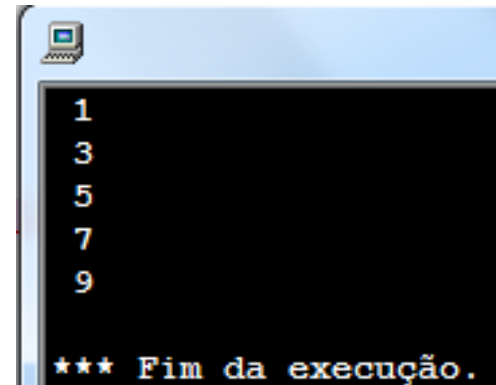
```
inicio
```

```
    Para n <- 1 ate 10 faca
```

```
        escreval(n)
```

```
    Fimpara
```

```
fimalgoritmo
```



```
1
3
5
7
9
*** Fim da execução.
```

Unidade 1

Estrutura de Repetição

Estrutura Para (for):

Exemplo, somar valores:

```
algoritmo "somador"  
  
var  
    n, c, s: inteiro  
inicio  
    s <- 0  
    Para c <- 1 ate 5 faca  
        escreval("Digite o valor: ")  
        Leia (n)  
        s <- s + n  
    Fimpara  
    Escreval ("A soma dos valores é :", s)  
fimalgoritmo
```

Unidade 1

Estrutura de Repetição

Estrutura **Para (for)**:

Exercício: digite 5 valores e verifique quantos estarão entre 0 e 10.

```
algoritmo "valores"
```

```
var
```

```
    c,v, tot10: inteiro
```

```
inicio
```

```
    tot10 <- 0
```

```
    Para c <- 1 ate 5 faca
```

```
        escreva("Digite o valor: ")
```

```
        Leia (v)
```

```
        se (v >= 0) e (v <= 10) entao
```

```
            tot10 <- tot10 + 1
```

```
        Fimse
```

```
    Fimpara
```

```
    Escreval ("Ao todo foram encontradas ", tot10," número(s) entre 0 e 10")
```

```
fimalgoritmo
```

Unidade 1

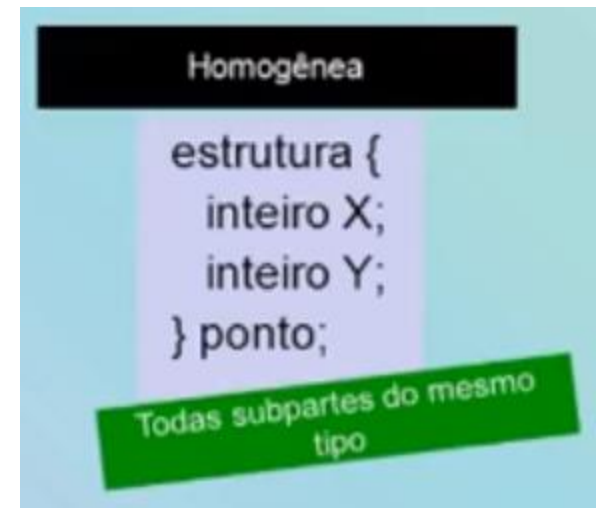
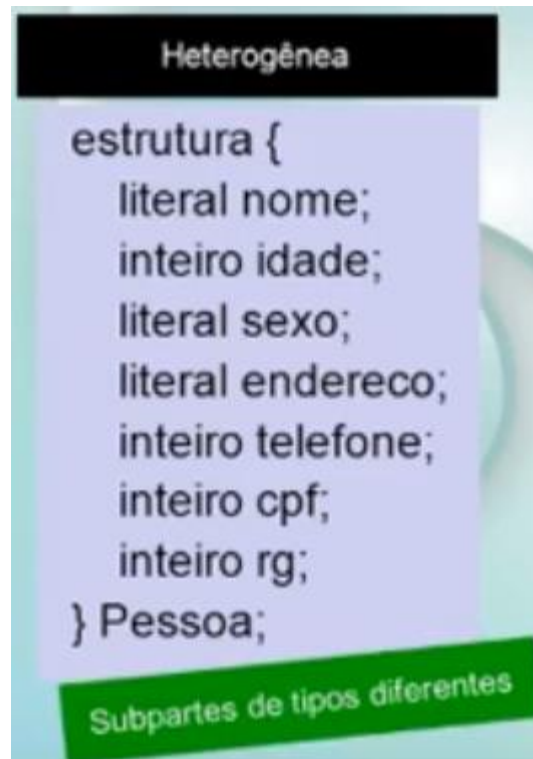
Estrutura de Dados

Homogenia – são do mesmo tipo

- Vetores e matrizes

Heterogenia

- Registros



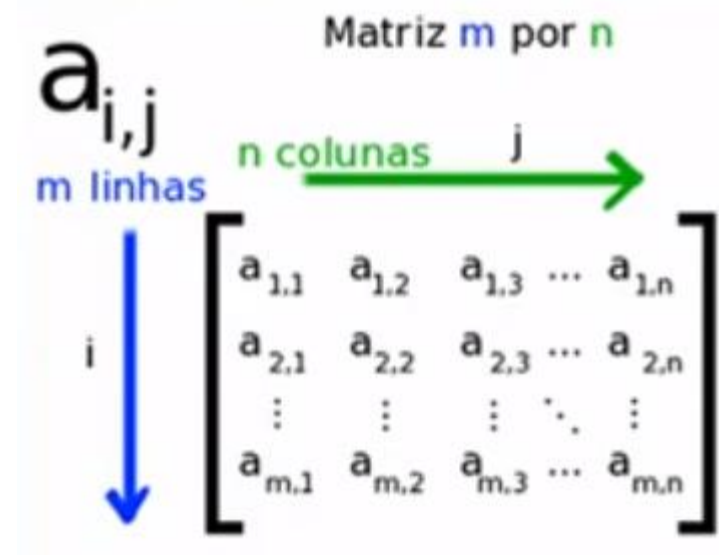
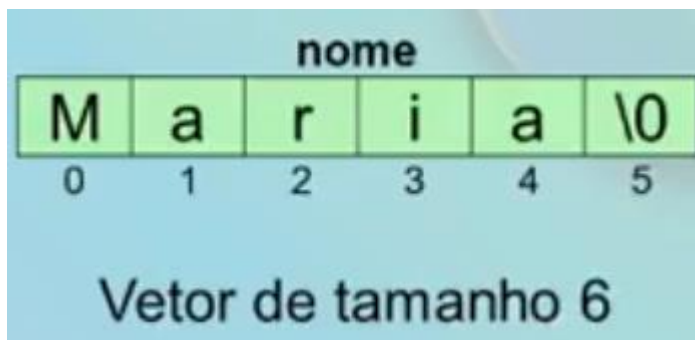
Unidade 1

Estrutura de Dados

Matrizes e vetores

Permitem armazenar uma coleção de variáveis do mesmo tipo.

Enquanto a **matriz** possui **varias dimensões**, o **vetor** possui apenas **uma dimensão**.



Unidade 1

Estrutura de Dados

Exemplo: Um programa para controlar notas de alunos, onde, preciso controlar 4 notas diferentes mais a frequência, tenho ao todo 60 alunos.

Posso gerar
vários vetores
ou então **uma**
matriz que
compõe tudo.

Vetor Notas 1						
A1	A2	A3	A4	A59	A60
Vetor Notas 2						
A1	A2	A3	A4	A59	A60
Vetor Notas 3						
A1	A2	A3	A4	A59	A60
Vetor Notas 4						
A1	A2	A3	A4	A59	A60
Vetor Falhas						
A1	A2	A3	A4	A59	A60

Unidade 1

Matriz

Os estudos sobre a **matriz** vêm desde o século XIX e traz uma nova experiência ao campo da matemática. Hoje, mesmo sem percebermos, o sistema matricial está envolvida a nossa volta, desde aos cálculos feito por um computador até a construção de **estruturas importantes** para o ser humano, sendo a base para computação gráfica, para trabalhar com jogos.

Unidade 1

Matriz

Para se entender matriz é importante observar primeiramente como as mesmas são formadas. Nas matrizes existem o que é chamamos de **linha** (os valores ordenados na horizontal) e o número delas é representado pela letra “m”. E o que chamamos de **coluna** (os valores ordenados na vertical), onde o número delas é representado pela letra “n”.

Um VETOR são variáveis Compostas com dimensões.

- ▶ Atribuir valores na declaração do vetor:

```
int vetor[] = new int[]{1,2,3,4,5};
```

- ▶ Atribuir valores na declaração da matriz:

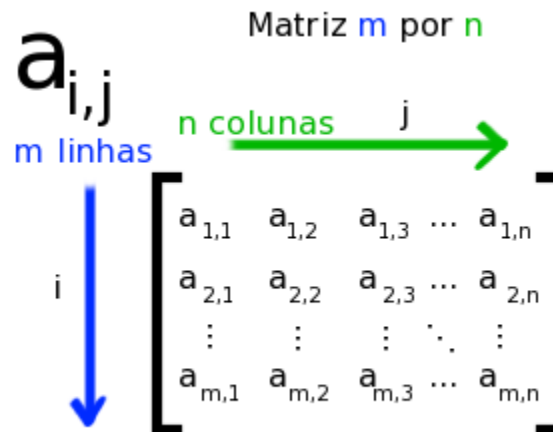
```
float matriz[][] = new float[][]{{1,2,3},{4,5,6}}
```

[Linha,Coluna]

Unidade 1

Matriz

Então se nos é dado uma sequencia de valores, eles devem ser convertidos de cima para baixo nas colunas e da esquerda para a direita nas linhas. Como mostram as setas azuis e verde na figura. Ex:



Unidade 1

Matriz

Vetor com 1 variável com 3 posições [índices]

```
algoritmo "Exercício_1"  
  // Função : Calcular 2 notas e mostrar a média  
  // Autor : Sandro  
  var  
    N: vetor[1..3] de real  
  inicio  
    N[1] <- 60  
    N[2] <- 80  
    N[3] <- ((N[1]+ N[2])/2)  
    Escreval("A média dos valores é: ", N[3])  
  
  fimalgoritmo
```

Unidade 1

Matriz

Exemplo:

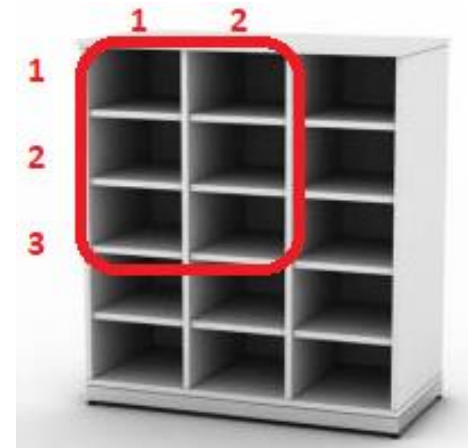
```
algoritmo "somar 5 valores"  
  
var  
    V: vetor [1..5] de inteiro  
    i: inteiro  
  
inicio  
    Para i <- 1 ate 5 faca  
        Escreva ( " Digite o ", i, "º número: ")  
        leia (V[i])  
    fimpara  
    i <- V[1] + V[2] + V[3] + V[4] + V[5]  
    escreva ("A soma dos números é :", i)  
  
fimalgoritmo
```

A variável i é atribuída 2 vezes uma no: para somatória

Escopo	Nome
GLOBAL	V[1]
GLOBAL	V[2]
GLOBAL	V[3]
GLOBAL	V[4]
GLOBAL	V[5]
GLOBAL	I

Unidade 1

Matriz



Vetor com 1 variável com 2 posições na horizontal e 3 na vertical:

```
algoritmo "semnome"
```

```
var
```

```
    n: vetor[1..3, 1..2] de inteiro
```

```
inicio
```

```
    n[1,1] <- 5
```

```
    n[1,2] <- 10
```

```
    n[2,1] <- 15
```

```
    n[2,2] <- 20
```

```
    n[3,1] <- 25
```

```
    n[3,2] <- 30
```

```
fimalgoritmo
```

[Linha,Coluna]

Unidade 1

Matriz

algoritmo "semnome"

var

n: vetor[1..3, 1..2] de inteiro
L, C: inteiro

inicio

para L <- 1 ate 3 faca
 para C <- 1 ate 2 faca
 leia (n[L,C])
 Fimpara
Fimpara

fimalgoritmo

	1	2
1		
2		
3		

Linha 1
Coluna 1

	1	2
1		
2		
3		

Linha 1
Coluna 2

	1	2
1		
2		
3		

Linha 2
Coluna 1

Escopo	Nome	Tipo	Valor
GLOBAL	N[1,1]	I	0
GLOBAL	N[1,2]	I	0
GLOBAL	N[2,1]	I	0
GLOBAL	N[2,2]	I	0
GLOBAL	N[3,1]	I	0
GLOBAL	N[3,2]	I	0
GLOBAL	L	I	1
GLOBAL	C	I	1

Unidade 1

Matriz

	A	B	C
1	86	16	67
2	46	51	23
3	60	38	27



```

Digite o valor da posição [ 1 , 1 ]86
Digite o valor da posição [ 1 , 2 ]16
Digite o valor da posição [ 1 , 3 ]67
Digite o valor da posição [ 2 , 1 ]46
Digite o valor da posição [ 2 , 2 ]51
Digite o valor da posição [ 2 , 3 ]23
Digite o valor da posição [ 3 , 1 ]60
Digite o valor da posição [ 3 , 2 ]38
Digite o valor da posição [ 3 , 3 ]27

*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.

```

algoritmo "matriz_num_par"

var

n: vetor[1..3, 1..3] de inteiro

L, C: inteiro

inicio

para L <- 1 ate 3 faca

para C <- 1 ate 3 faca

Escreva ("Digite o valor da posição [", L, " , ", C, "]")

leia (n[L,C])

Fimpara

Fimpara

fimalgoritmo

Unidade 1

Matriz

algoritmo "matriz_num_par"

var

n: vetor[1..3, 1..3] de inteiro

L, C: inteiro

inicio

```
para L <- 1 ate 3 faca
  para C <- 1 ate 3 faca
    Escreva ("Digite o valor da posição [", L, " , ", C, " ]")
    leia ( n[L,C])
```

Fimpara

Fimpara

```
para L <- 1 ate 3 faca
  para C <- 1 ate 3 faca
    Se (n[L,C] % 2 = 0) entao
      Escreva ( "(" , n[L,C]: 2, ")")
    senao
```

Escreva (n[L, C]: 4)

fimse

Fimpara

Escreval ()

Fimpara

fimalgoritmo

```
Digite o valor da posição [ 1 , 1 ]71
Digite o valor da posição [ 1 , 2 ]18
Digite o valor da posição [ 1 , 3 ]81
Digite o valor da posição [ 2 , 1 ]45
Digite o valor da posição [ 2 , 2 ]49
Digite o valor da posição [ 2 , 3 ]75
Digite o valor da posição [ 3 , 1 ]20
Digite o valor da posição [ 3 , 2 ]100
Digite o valor da posição [ 3 , 3 ]13
```

```
71(18) 81
45 49 75
(20)(100) 13
```

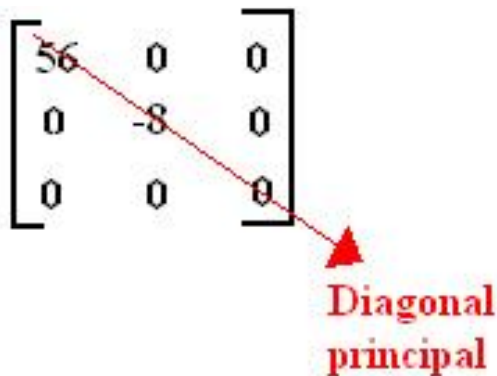
espaço



Unidade 1

Matriz Diagonal Principal

Para que uma matriz tenha **diagonal** ela deverá ser uma matriz quadrada, então uma matriz diagonal é uma matriz quadrada onde os elementos que não pertencem à diagonal principal são obrigatoriamente iguais a zero.

$$A = \begin{bmatrix} 56 & 0 & 0 \\ 0 & -8 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$


Diagonal principal

Unidade 1

Matriz Diagonal Principal

A **matriz identidade** ou chamada também de matriz unidade é uma matriz quadrada de ordem n sendo que $n \geq 2$, onde os elementos que pertencem à diagonal principal são sempre iguais a 1 e os outros elementos que não pertencem à diagonal principal são iguais a zero.

$$I_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}_{3 \times 3}$$

Unidade 1

Matriz Diagonal Principal

```
algoritmo "matriz_identidade"

var
  n: vetor[1..4, 1..4] de inteiro
  L, C: inteiro
inicio

  para L <- 1 ate 4 faca
    para C <- 1 ate 4 faca
      Se ( L = C ) entao
        n[L,C] <- 1
      senao
        n[L,C] <- 0
      Fimse
    Fimpara
  Fimpara
  para L <- 1 ate 4 faca
    para C <- 1 ate 4 faca
      Escreva ( n[L, C]: 4)
    fimpara
  Escreval ()
Fimpara

fimalgoritmo
```

Exemplo – matriz de identidade 4º ordem

1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

*** Fim da execução.

*** Feche esta janela para retornar ao Visualg.

Unidade 1

Matriz

Exemplo – soma Diagonal Principal

```
algoritmo "Soma_Diagonal_Principal"

var
  n: vetor[1..4, 1..4] de inteiro
  L, C, SomaDiagonal: inteiro
inicio
  SomaDiagonal <- 0
  para L <- 1 ate 4 faca
    para C <- 1 ate 4 faca
      Escreva ("Digite o valor da posição [", L, " , ", C, " ]")
      leia ( n[L,C])
      Se ( L = C ) entao
        SomaDiagonal <- SomaDiagonal + n[L,C]
      Fimse
    Fimpara
  Fimpara
  para L <- 1 ate 4 faca
    para C <- 1 ate 4 faca
      Escreva ( n[L, C]: 4)
    Fimpara
  Escreval ()
Fimpara
Escreval (" A soma da Diagonal Principal é: " , SomaDiagonal)

fimalgoritmo
```

Unidade 1

Estrutura de Dados

Estrutura de dados Classificação:

- Lineares e não lineares

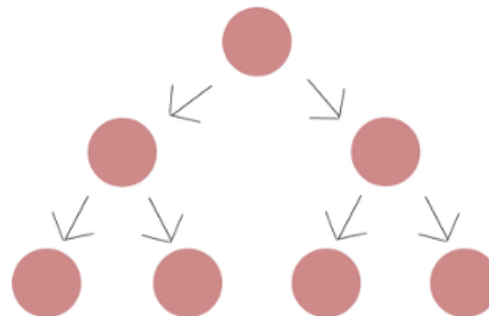
Estruturas de dados **Lineares**:

- Listas, Pilhas e Filas



Estruturas de dados **não Lineares**:

- Árvore e Grafo

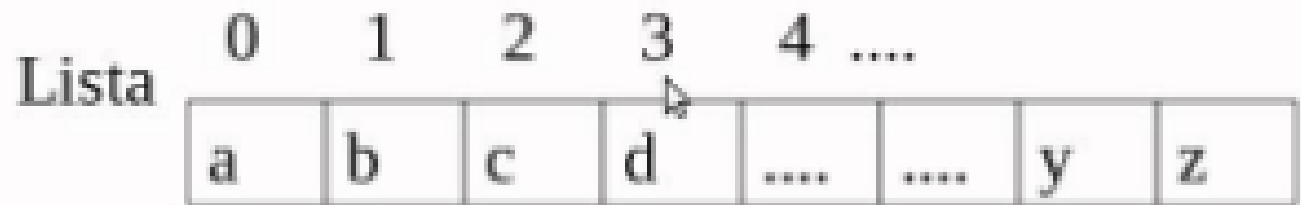


“Estas estruturas mostram como os dados são manipulados”

Unidade 1

Estrutura de Dados

Lista – São estruturas que permitem representar um conjunto de dados, que de alguma forma se relacionam, de forma que os elementos fiquem dispostos em **sequencia**.



Unidade 1

Estrutura de Dados

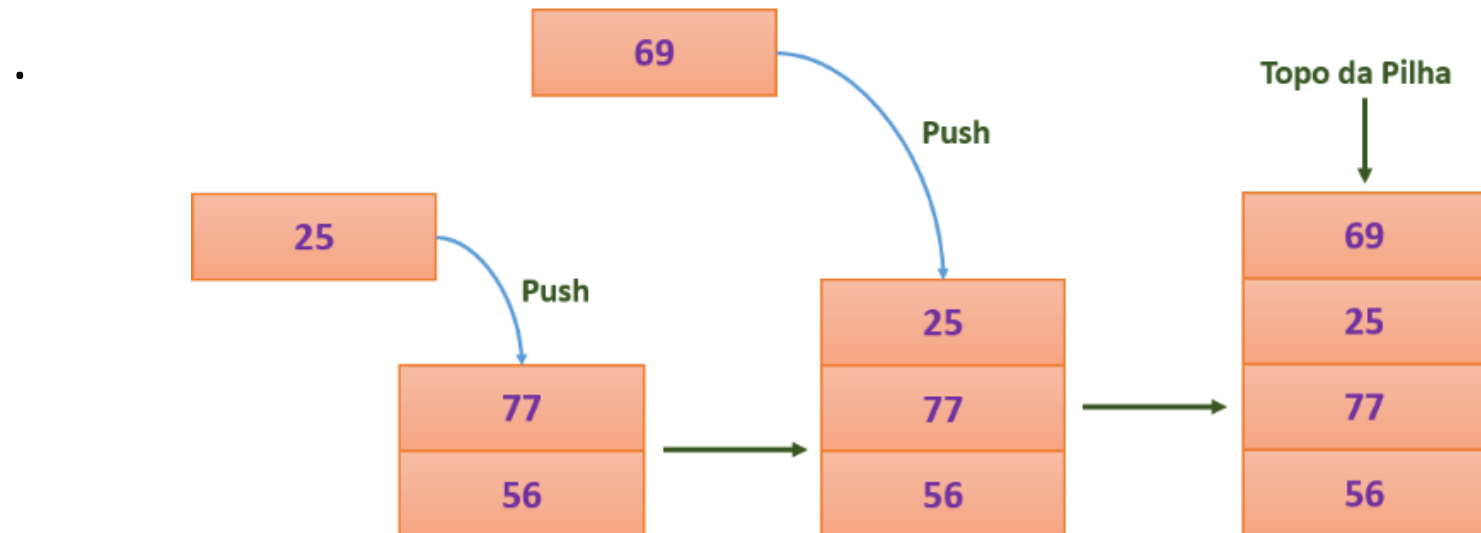
Lista – Operações mais comuns

- Criar
- Verificar lista vazia
- Verificar lista cheia
- Inserir
- Alterar
- Remover
-
- Exibir a quantidade
- Combinar
- Dividir lista
- Ordenar
- Esvaziar

Unidade 1

Estrutura de Dados

Pilha – É um tipo especial de lista onde os elementos a serem inseridos ou removidos ocorrem no **topo da pilha**. Esta característica é conhecida como LIFO (Last in, First Out- Último a entrar, primeiro a sair).



Unidade 1

Estrutura de Dados

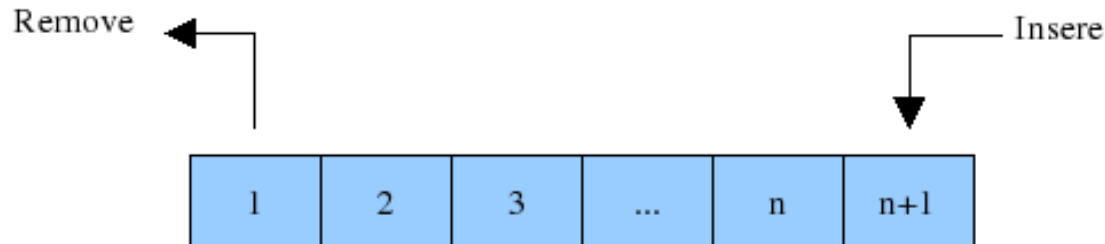
Pilha – Operações mais comuns

- Criar – cria uma pilha vazia
- Empilhar – Insere um novo elemento no topo
- Desempilhar – remove um elemento do topo da pilha
- Exibir topo ou a quantidade
- Esvaziar – esvazia todos os elementos da pilha.
- .

Unidade 1

Estrutura de Dados

Fila – É um tipo especial de lista onde os elementos a serem inseridos em uma extremidade, chamado de **início da fila**, e **final da fila**. Esta característica é conhecida como FIFO (First in, First Out- Primeiro a entrar, primeiro a sair).



Unidade 1

Estrutura de Dados

Fila – Operações mais comuns

Criar – cria uma fila vazia

Enfileirar – Insere um novo elemento no fim da fila

Desenfileirar – remove um elemento no início da fila

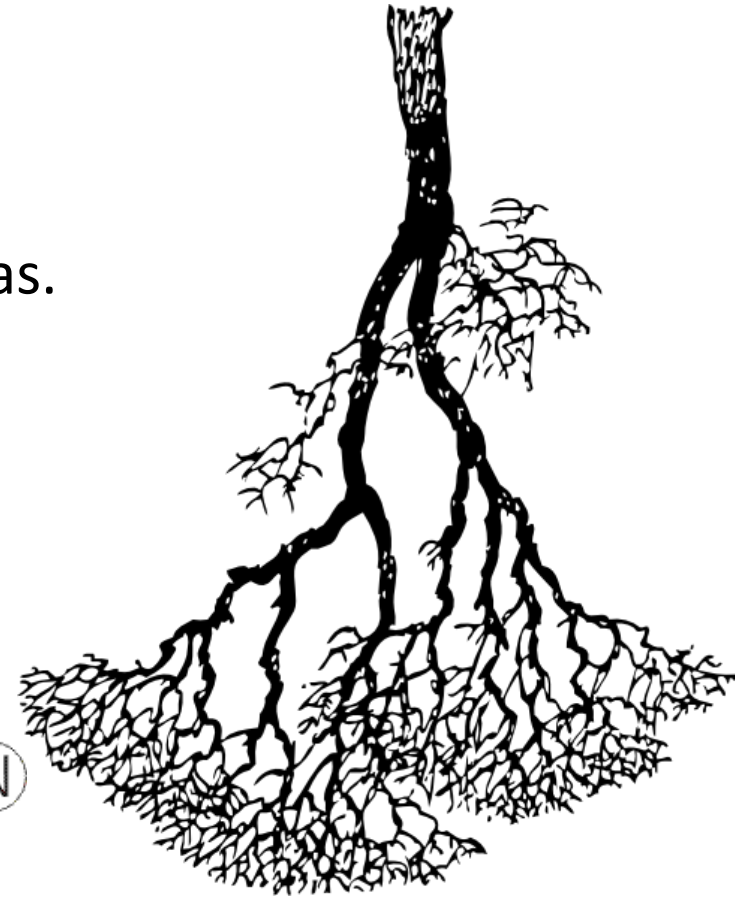
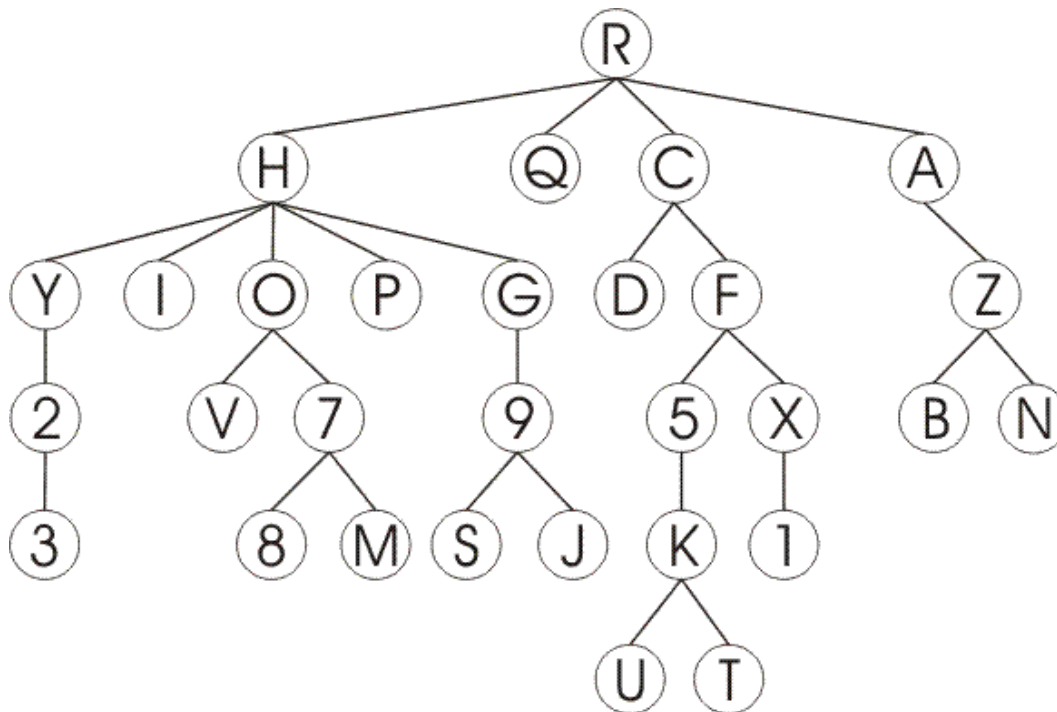
Exibir a quantidade – retorna a quantidade de elementos na fila.

Esvaziar – esvazia a fila.

Unidade 1

Estrutura de Dados

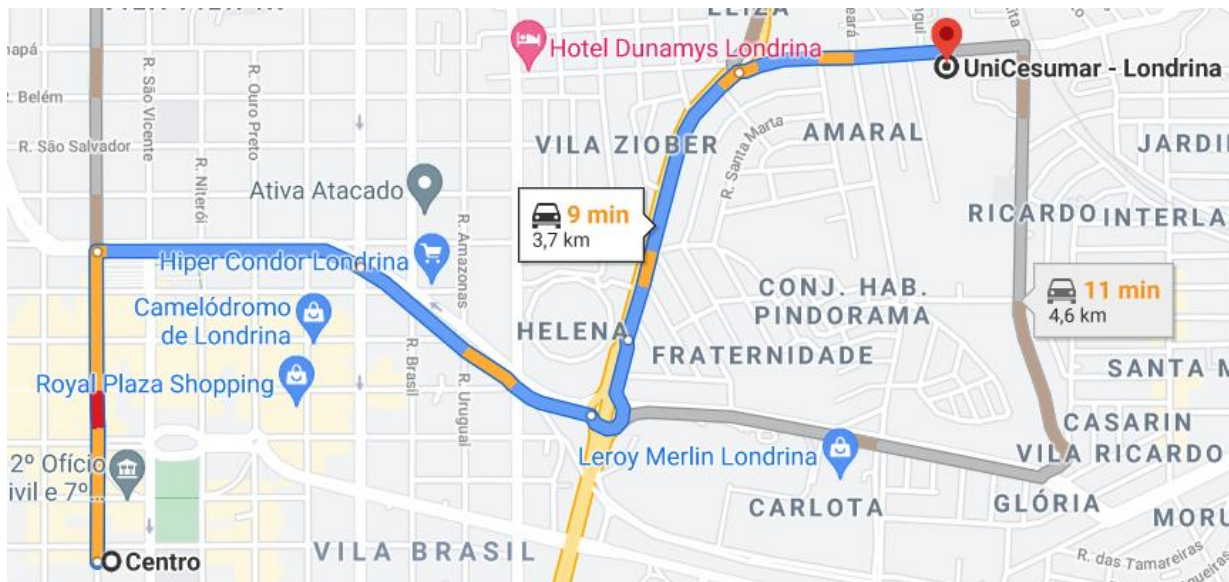
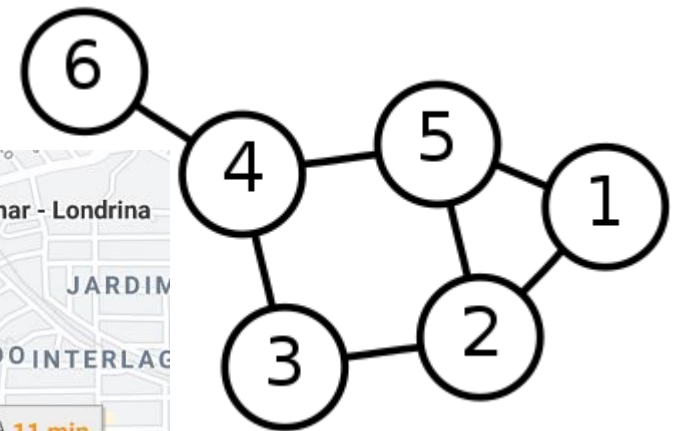
Árvores – É composta de nós e arestas.



Unidade 1

Estrutura de Dados


Grafos – relação entre os objetos de um determinado conjunto, com nós e arestas.



Unidade 1

Questão 1


Assinale a opção correta em relação a estrutura de dados pilha.

- a) O primeiro elemento a ser inserido será o primeiro a ser retirado.
- b) O primeiro elemento a ser inserido irá até o fim da sequência e será retirado.
- c) O elemento é retirado da base da pilha
-  d) O último elemento a ser inserido será o primeiro elemento a ser retirado.
- e) O último elemento a ser inserido será o último elemento a ser retirado.

Unidade 1

Questão 2

Qual a melhor estrutura de dados para representar problemas do tipo de redes de computadores, trajetos entre cidades e roteamento de veículos, etc?

- a) Lista
- b) Pilha
- c) Fila
- d) Árvore
-  e) Grafo

Unidade 1



Referências:

SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. Estruturas de dados e seus algoritmos. 2ed. Rio de Janeiro: LTC, 1994. 320p.

TENENBAUM, Aaron M.; LANGSAM, Yedidiah; AUGENSTEIN, Moshé J.. Estruturas de dados usando C. São Paulo: Makron Books, 1995. 884p.

VELOSO, Paulo et al.. Estruturas de dados. Rio de Janeiro: Campus, 2001. 228p

Atividades - Unidade 1

