PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS

ESCOLA POLITÉCNICA E DE ARTES

ESTRUTURA DE DADOS ORIENTADA A OBJETOS

ADS1232

PROF. MSC. ANIBAL SANTOS JUKEMURA

Conceitos Gerais

PUC GOIÁS

"Na resolução de um problema por meio de um programa, a primeira providência é conceber um algoritmo adequado."

Extraído de : Estruturas de Dados – Homero L. Pícollo [2].

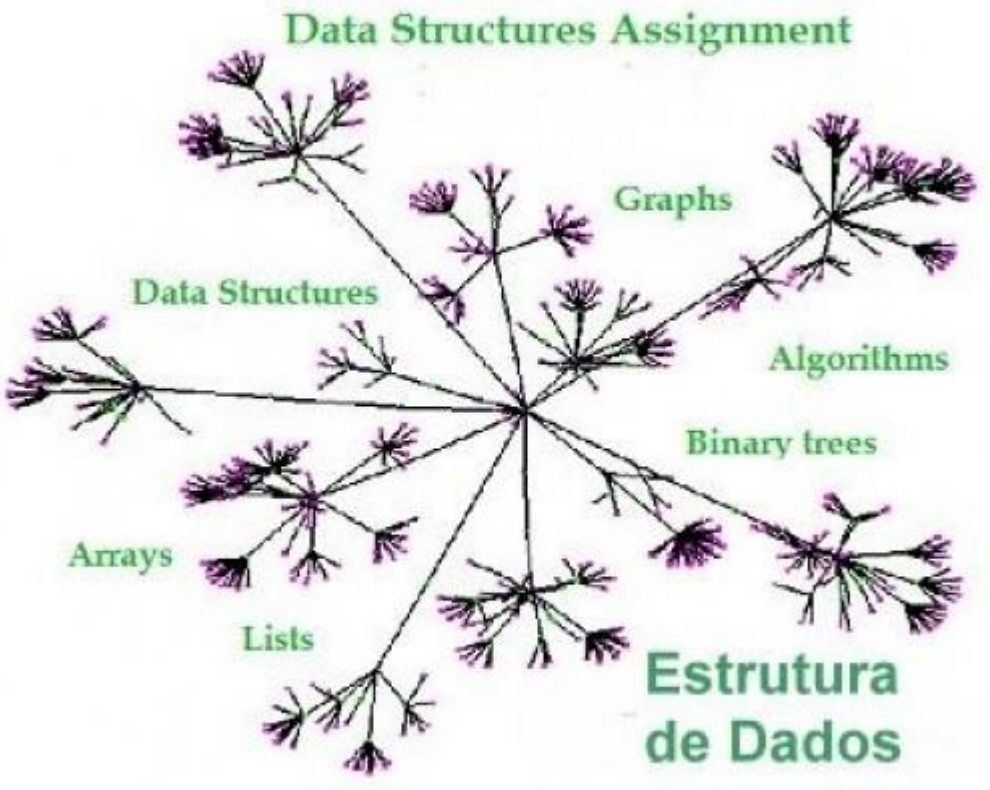
- A eficiência de um algoritmo qualquer está intimamente relacionada à disposição, na memória, dos dados que são tratados pelo programa.

- Por exemplo, se freqüentemente enfrentamos o problema de descobrir os números de telefones de nossos conhecidos, é conveniente dispor de uma relação de números, organizada em uma agenda.

- Se a organização for feita por ordem alfabética, a agenda de fato ajuda. Se, porém, organizássemos nossa agenda pela ordem de altura das pessoas, com raras exceções, a agenda se tornaria difícil de manusear.

PUC GOIÁS

As estruturas de dados são formas de distribuir e relacionar os dados disponíveis, de modo a tornar mais eficientes os algoritmos que manipulam esses dados.



PUC GOIÁS

Problema: Manipular um conjunto de fichas de um fichário.

Solução: **Organizar** as fichas em ordem alfabética.

Operações: possíveis Inserir ou retirar um ficha, procurar uma ficha, etc.

Solução Computacional ?

Estrutura de Dados Correspondente:
LISTA – seqüência de elementos dispostos em ordem.

PUC GOIÁS

Problema:
Organizar as pessoas que querem ser atendidas num guichê.

Solução: Colocar as pessoas em **FILA**.

Operações possíveis À medida que uma pessoa é atendida no guichê, outra entra no final da fila. Não é permitido “furar” a fila, ou seja, entrar uma pessoa entre outras que já estão presentes.

Solução Computacional ?

Estrutura de Dados Correspondente:
FILA - Seqüência de elementos dispostos em ordem com uma regra para a entrada e saída dos elementos - (o 1º que chega também é o 1º que sai da estrutura).

PUC GOIÁS

Problema: Organizar um conjunto de pratos que estão sendo lavados, um a um, em um restaurante.

Solução: Colocar os pratos empilhados.

Operações possíveis: Colocar um prato limpo no alto da pilha, retirar um prato do alto da pilha, etc.

Solução Computacional ?

Estrutura de Dados Correspondente:
PILHA – seqüência de elementos dispostos em ordem, mas com uma regra para entrada e saída dos elementos (o último que chega é o primeiro que sai da estrutura).

PUC GOIÁS

Problema: Conseguir um modo de visualizar o conjunto de pessoas que trabalham em uma empresa, tendo em conta sua função.

Solução: Construir um organograma da empresa.

Operações possíveis: Inserir ou retirar certas funções, localizar uma pessoa, etc.

Solução Computacional ?

Estrutura de Dados Correspondente:
ÁRVORE – estrutura de dados que caracteriza uma relação de hierarquia entre os elementos (uma pessoa não pode pertencer a dois deptos. diferentes, cada diretoria tem os seus próprios deptos., etc.).

PUC GOIÁS

Problema:
Estabelecer um trajeto para percorrer todas as capitais de um país.

Solução: Utilizar um mapa que indique as rodovias existentes e estabelecer uma ordem possível para percorrer todas as cidades.

Operações possíveis: Encontrar um modo de percorrer todas as cidades, determinar o caminho mais curto para ir de uma cidade para outra, etc.

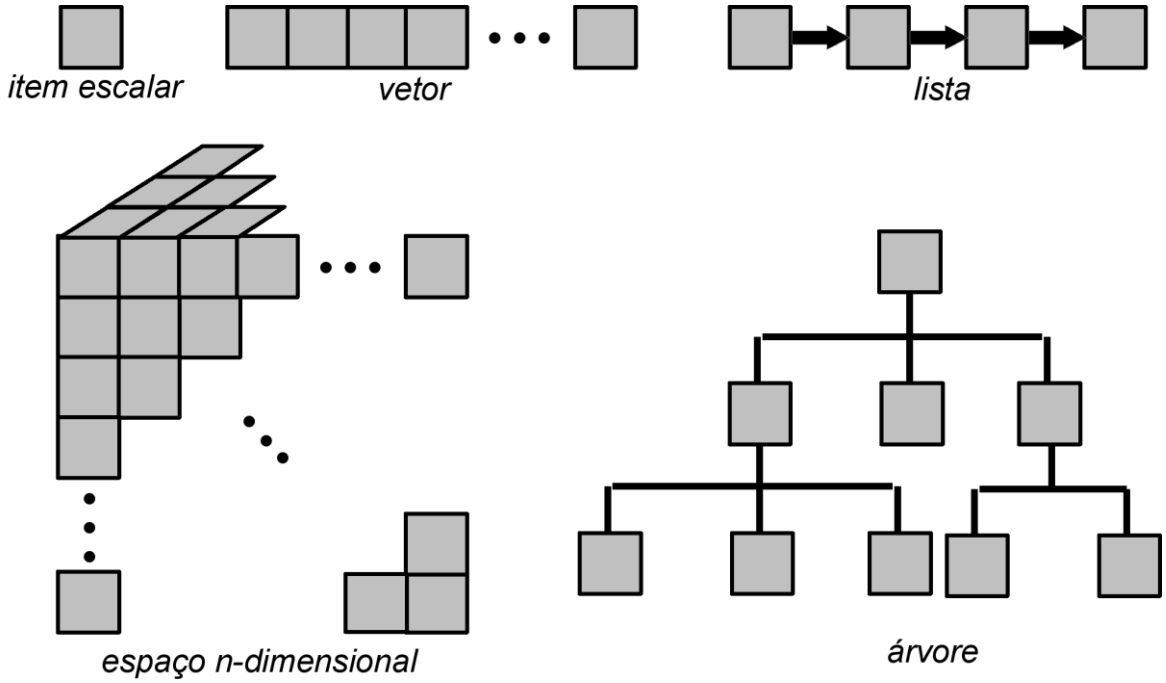
Solução Computacional ?

Estrutura de Dados Correspondente:
GRAFO – estrutura bastante genérica que organiza vários elementos, estabelecendo relações entre eles, dois a dois.

PUC GOIÁS

Os exemplos vistos correspondem exatamente aos tipos básicos de estruturas de dados utilizadas em computação:

Listas, **Filas**, **Pilhas**, **Árvores** e **Grafos**.



PUC GOIÁS

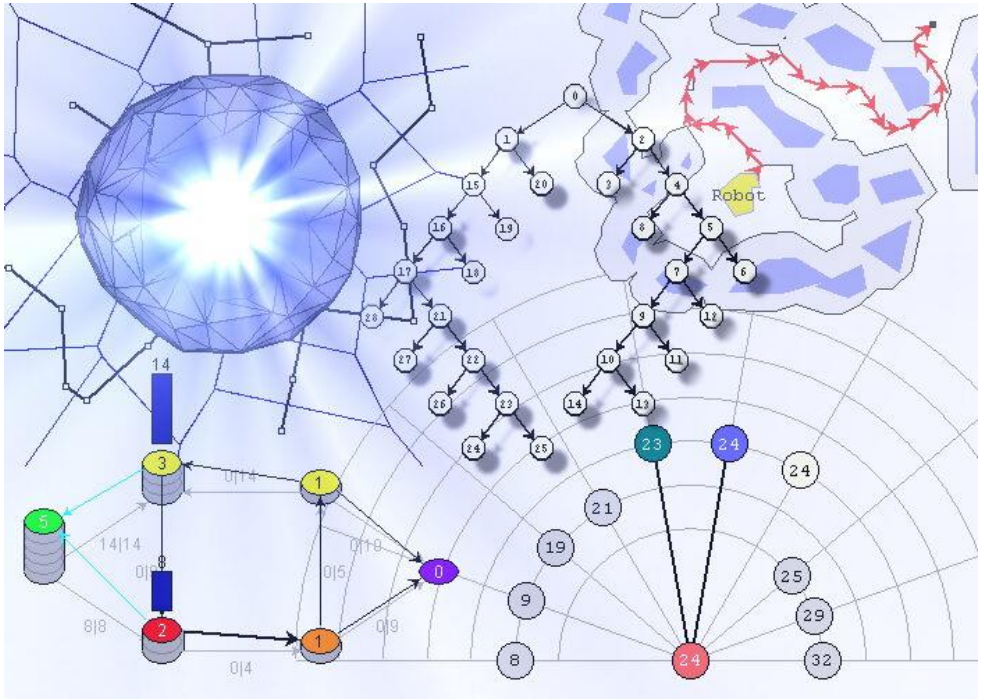
- O estudo de estrutura de dados é parte fundamental para o desenvolvimento de programas e algoritmos.

- Assim como um número pode ser representado em vários sistemas diferentes, também um conjunto de dados relacionados entre si pode ser descrito através de várias estruturas de dados distintas.

- Quando o programador cria um algoritmo para solucionar um problema, ele também cria uma **estrutura de dados** que é manipulada pelo algoritmo.

PUC GOIÁS

A escolha de uma determinada estrutura **pode afetar substancialmente** a quantidade de área de armazenamento requerida para o processamento bem como o tempo deste processamento.



Um programador de pouca experiência utilizará em sua programação formas bastante simplificadas para a representação dos dados envolvidos (como matrizes e vetores), que possivelmente seriam **adequadamente representados** através de **estruturas mais sofisticadas** (filas encadeadas, árvores, etc.), tornando o processamento mais eficiente em termos de área de armazenamento e tempo.

PUC GOIÁS

Estrutura de dados: o que é?

Uma estrutura de dados pode ser dividida em dois pilares fundamentais: ***dado*** e ***estrutura***

Dado

Elemento que possui valor agregado e que pode ser utilizado para solucionar problemas computacionais. Os dados possuem tipos específicos.

Estrutura

Elemento estrutural que responsável por carregar as informações dentro de uma estrutura de *software*.

PUC GOIÁS

Estrutura de dados: o que é?

Uma estrutura de dados pode ser dividida em dois pilares fundamentais: ***dado*** e ***estrutura***

Dado

Tipos de dados:
- Inteiro (int)
- Texto (string)
- Caracter (char)
- Ponto flutuante (float)
- Ponto flutuante (double)

Estrutura

Estruturas:
- Vetores multidimensionais
- Pilhas
- Filas
- Listas

PUC GOIÁS

Tipos de Dados

Um algoritmo, basicamente, consiste na entrada de dados, seu processamento e a saída dos dados computados. Existem dados considerados básicos.

São eles:

- **Dados numéricos** : armazenam valores numéricos.

- **Dados alfanuméricos** : armazenam valores alfabéticos, numéricos, sinais, etc...

- **Dados Lógicos** : armazenam valores lógicos.

PUC GOIÁS

Tipos de Dados

- Em muitos casos, a utilização dos dados básicos pode satisfazer o requisito de uma aplicação.

- Em outros casos, pode-se utilizar um tipo de dados um pouco mais sofisticado, do tipo matrizes e vetores.

PUC GOIÁS

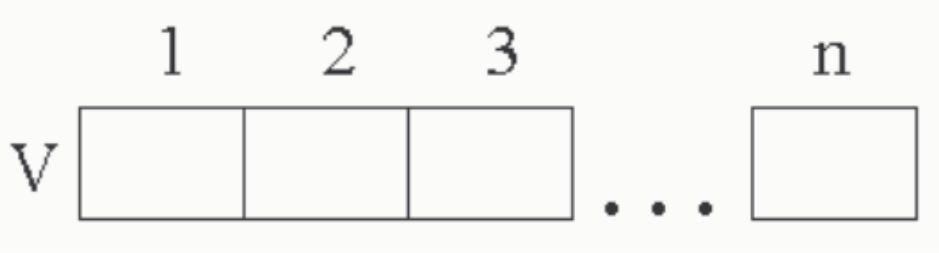
Tipos de Dados

- **Vetores** : um arranjo unidimensional de dados do tipo básico, com um mesmo identificador (nome), mas diferenciado pela sua posição (índice) dentro do vetor.

  Pode-se definir um vetor como:

$$V = V(1), V(2), ..., V(n)$$

onde *V(i)* é o i-ésimo elemento do vetor *V* e $1 <= i <= n$.



PUC GOIÁS

Tipos de Dados

Principais tipos de estruturas de dados

Vetores (unidimensionais e bidimensionais)

É uma das estruturas de dados mais simples e mais utilizadas dentre todas. Principais características:

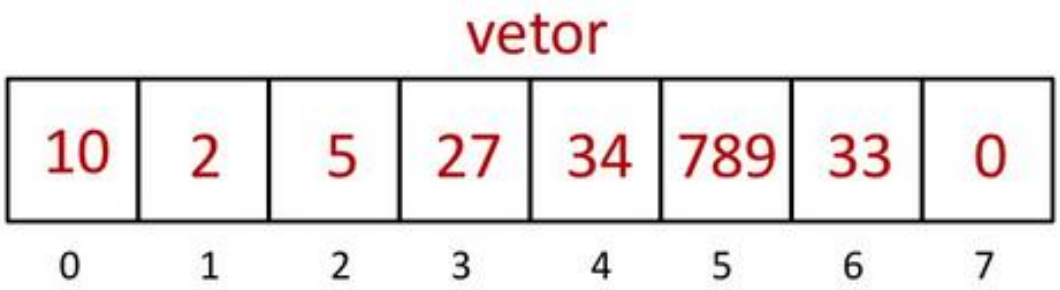
- *Adição e pesquisa de novos elementos de forma aleatória*
- *Acesso aos elementos através de índices*
- *Possuem tamanho finito de elementos*
- *Carregam dados de tipos específicos*
- *Indexação com início em 0 (zero)*
- *Unidimensionais: apenas linha*
- *Bidimensionais: linhas e colunas (formato de matriz)*

PUC GOIÁS

Tipos de Dados

Principais tipos de estruturas de dados

Vetores unidimensionais (int vetor[8];**)**



- vetor[0] = 10;
- vetor[1] = 2;
- vetor[2] = 5;
- vetor[3] = 27;
- vetor[4] = 34;
- vetor[5] = 789;
- vetor[6] = 33;
- vetor[7] = 0;

PUC GOIÁS

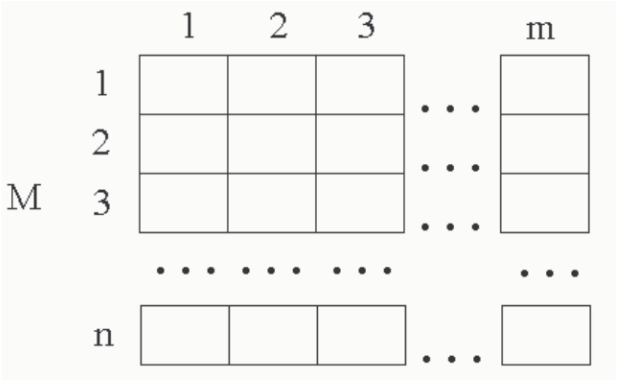
Tipos de Dados

- **Matrizes** : um arranjo de várias dimensões de dados do tipo básico, com um mesmo identificador (nome), mas diferenciado pelas suas posições (índices) dentro da matriz.

  Uma matriz bidimensional pode ser definida como:

$$M = \begin{matrix} M(1,1) & M(1,2) & ,,, & M(1,m) \\ M(2,1) & M(2,2) & \ldots & M(2,m) \\ \ldots & \ldots & \ldots & \ldots \\ M(n,1) & M(n,2) & \ldots & M(n,m) \end{matrix}$$

onde *M(i,j)* indica o elemento da matriz *M* posicionado na i-ésima linha e j-ésima coluna, sendo que 1 <= *i* <= *n* e 1 <= *j* <= *m*.



PUC GOIÁS

Tipos de Dados

Principais tipos de estruturas de dados

Vetores bidimensionais (int vetorb[2][2];**)**

- vetorb[0][0] = 10;
- vetorb[0][1] = 2;
- vetorb[1][0] = 34;
- vetorb[1][1] = 50;

	0	1
0	10	2
1	34	50

vetorb

PUC GOIÁS

Nodo (Nó)

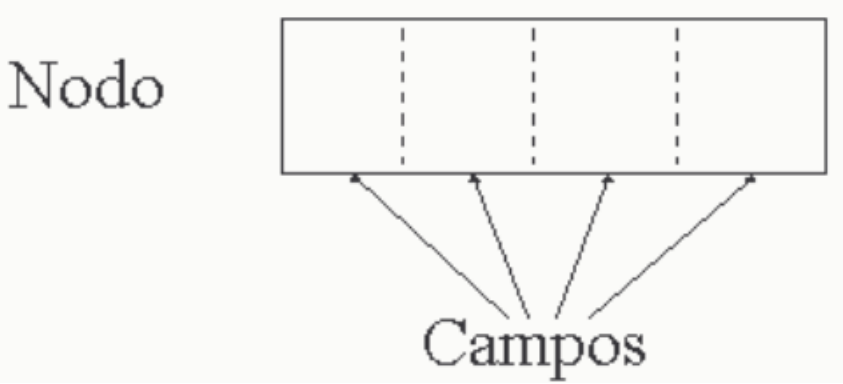
- O termo estrutura de dados é utilizado para referenciar diferentes formas de representação de dados.

- A escolha de uma determinada estrutura para representar um conjunto de dados relacionados deve-se, principalmente, ao tipo de operações que serão realizadas sobre o mesmo, usando uma manipulação otimizada em relação ao tempo necessário para efetuar as operações e a área de armazenamento requisitada para guardar estes dados.

- A entidade elementar de uma estrutura de dados é o nodo (nó). Um nodo é diferenciado pelo seu endereço relativo dentro da estrutura e pode ser constituído de um ou vários campos.

PUC GOIÁS

Nodo (Nó)



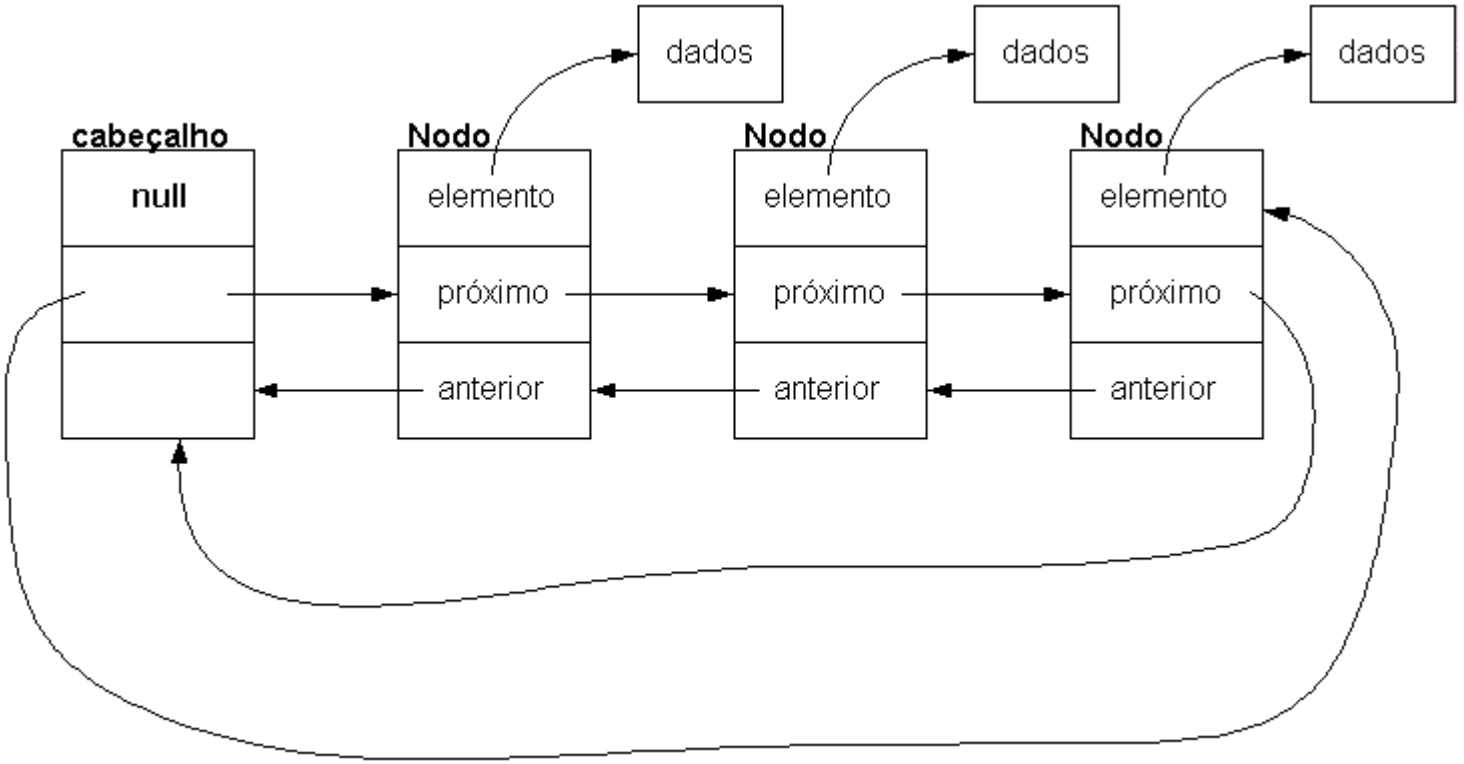
- Cada campo de um nodo armazena uma informação, um dado, que pode ser do tipo numérico, alfanumérico, lógicos, imagens, sons, enfim, qualquer informação que possa ser manipulada.

- Todos os nodos de uma mesma estrutura devem ter a mesma configuração.

PUC GOIÁS

Nodo (Nó)

Exemplo:



PUC GOIÁS

Listas Lineares

- Uma lista é uma sequência ordenada de elementos do mesmo tipo. Por exemplo, um conjunto de fichas de clientes de uma loja, organizadas pela ordem alfabética dos nomes dos clientes.
- Neste fichário é possível introduzir uma nova ficha ou retirar uma velha, alterar os dados de um cliente etc. 6
- Do ponto de vista matemático, uma lista é uma sequência de zero ou mais elementos de um determinado tipo. Geralmente se representa uma lista de elementos, separando-os por vírgulas.

a1 , a2 , ... Na

onde n >= 0 e ai é um elemento da lista. O número n é dito comprimento da lista. Se n=0 temos uma lista vazia.

PUC GOIÁS

Listas Lineares

Definição dada por Knuth para uma lista linear :

“Uma lista linear X é um conjunto de nodos X(1), X(2), ...,X(n), tais que:

a) X(1) é o primeiro nodo da lista;
b) X(n) é o último nodo da lista; e
c) Para 1<k”

PUC GOIÁS

Listas Lineares

Basicamente, o manuseio de listas lineares envolve três tipos de operações:

1) Inserção de novos nodos à lista;
2) Retirada de nodos da lista;
3) Consulta de conteúdo de algum nodo da lista.

PUC GOIÁS

Listas Lineares

Naturalmente existem outras operações que podem ser efetuadas sobre uma lista, que não são tão importantes para a caracterização dos diferentes tipos de listas, tais como:

4) Concatenar duas listas;
5) Determinar o número de nodos de uma lista;
6) Localizar um nodo da lista com um determinado conteúdo;
7) Criação de uma lista;
8) Cópia de uma lista;
9) Ordenação de uma lista por algum critério;
10) Destruição de uma lista;
11) Modificação do conteúdo de algum nodo da lista.

PUC GOIÁS

Listas Lineares

- As modalidades mais utilizadas de listas lineares são aquelas em que as três operações (1, 2 e 3) são realizadas nas extremidades da lista. Além disso, ao se manipular listas lineares, as operações frequentemente se repetem.

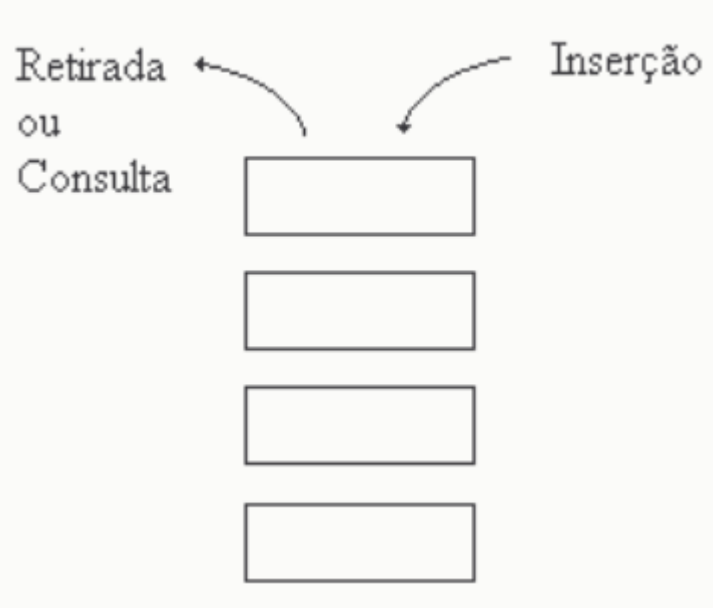
- Devido à constante utilização de certas formas básicas de operações, foram definidos alguns tipos de listas lineares clássicas, de acordo com a forma que essas operações são realizadas.

- Estes tipos particulares de listas lineares são: **PILHAS** e **FILAS**.

PUC GOIÁS

Listas Lineares

- **PILHA (Stack) :** é uma lista linear em que todas as operações (inserção, retirada e consulta) são realizadas numa única extremidade da estrutura. Graficamente, uma pilha é representada por:



PUC GOIÁS

Listas Lineares

Pilha

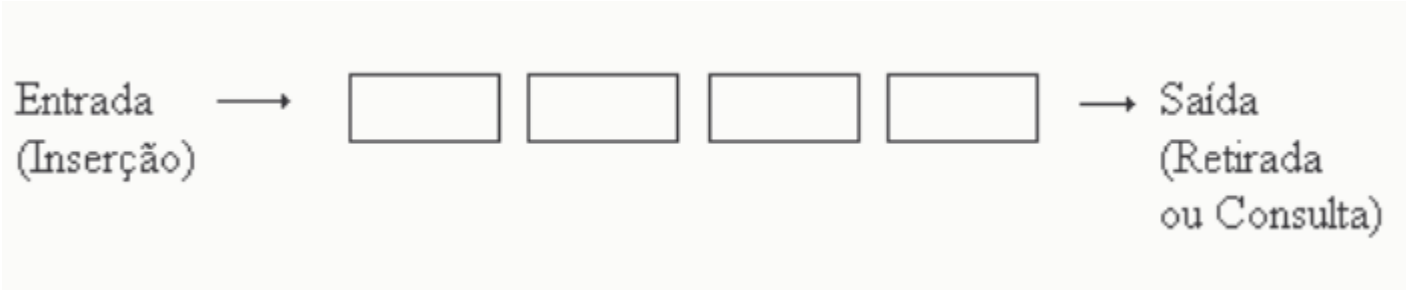
É uma estrutura de dados amplamente utilizada e que implementa a ideia de pilha de elementos:

- *LIFO (Last-In-First-Out)*
- *Permite a adição e remoção de elementos*
- *O elemento a ser removido é sempre aquele mais novo*
- *Simula a ideia de pilhas de elementos*
- *Para que o acesso a um elemento da pilha ocorra, os demais acima devem ser removidos*

PUC GOIÁS

Listas Lineares

- **FILA (Queue) :** é uma lista linear em que as inserções de novos nodos na estrutura são realizadas em uma das extremidades e as retiradas ou consultas aos nodos são realizadas na outra extremidade da lista. Graficamente tem-se:



PUC GOIÁS

Listas Lineares

FILA:

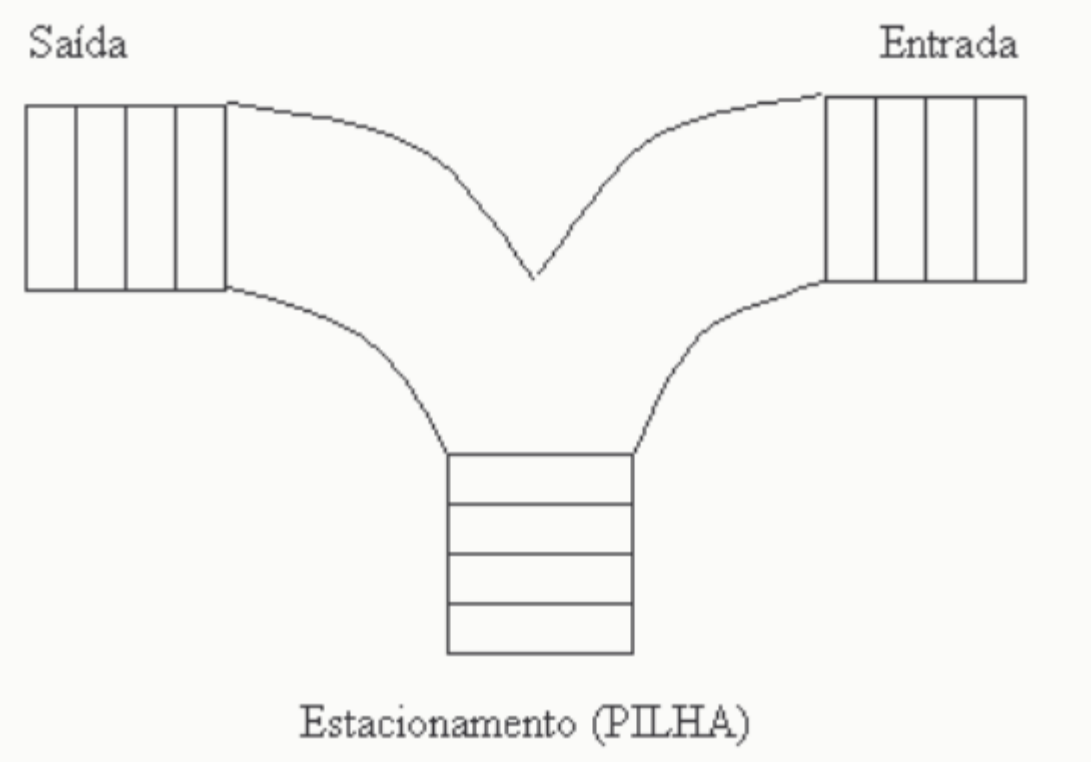
É uma estrutura de dados amplamente utilizada e que implementa a ideia de lista de elementos:

- *FIFO (First-In-First-Out)*
- *Permite a adição e remoção de elementos*
- *O elemento a ser removido é sempre o primeiro a entrar*
- *As operações de entrada e saída sempre ocorrem nas extremidades*

PUC GOIÁS

Listas Lineares

- **Exercício:** Imagine um terminal férreo com a seguinte configuração de trilhos:



PUC GOIÁS

Listas Lineares

- **Exercício:** Imagine um terminal férreo com a seguinte configuração de trilhos:

Os vagões só podem entrar na área Estacionamento (que é estruturado como uma pilha) vindos da Entrada e só podem sair da pilha pela Saída.

Denotando-se por I a entrada de um vagão no Estacionamento (inserção) e por R a saída de um vagão do Estacionamento (retirada) e considerando-se que na Entrada há quatro vagões numerados de 1 a 4, respectivamente (1 2 3 4) a execução da seqüência de operações de inserções e retiradas

I I R I I R R R

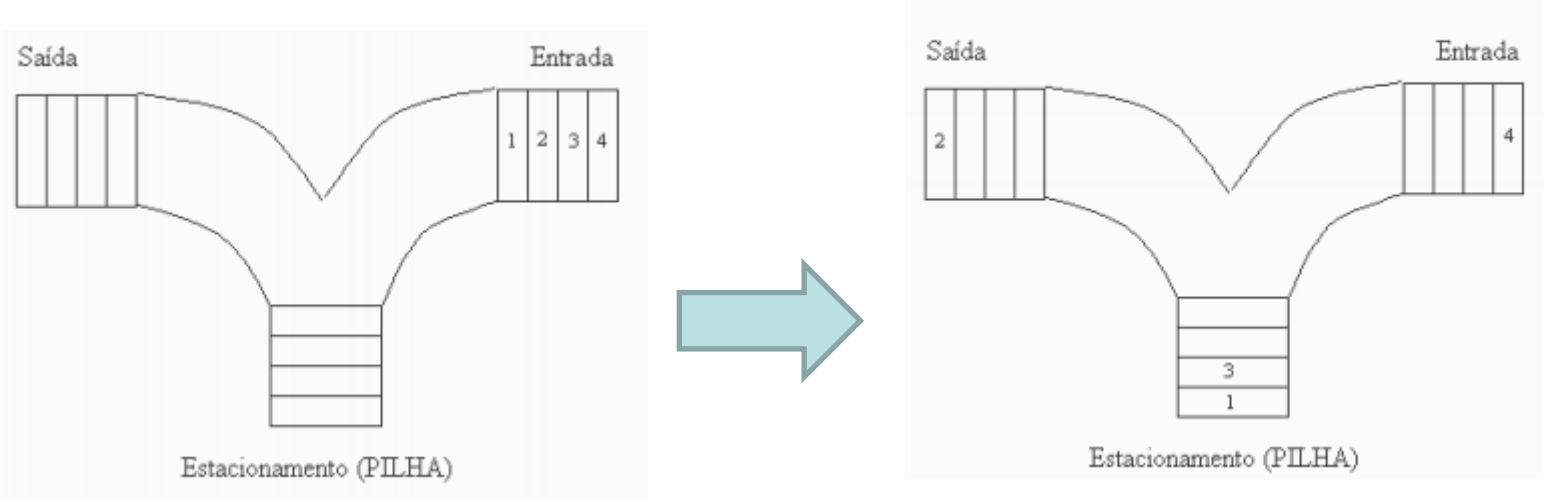
sobre os vagões da Entrada resultará em qual permutação dos vagões na Saída?

PUC GOIÁS

Listas Lineares

- **Exercício:** resposta
- (a), após a execução da seqüência **I I R I**, obter-se-á:

(a)

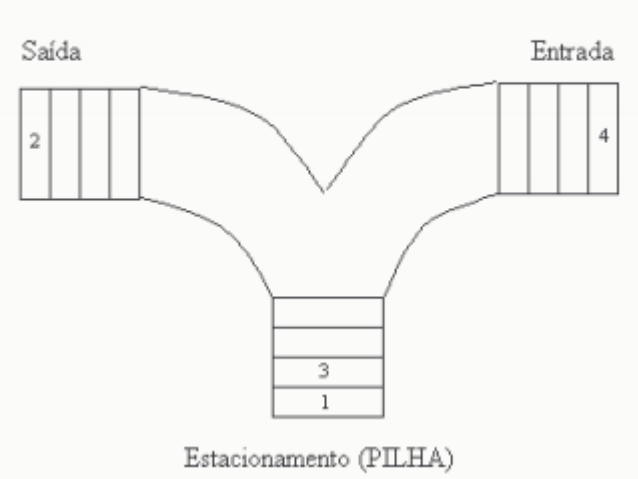


Listas Lineares

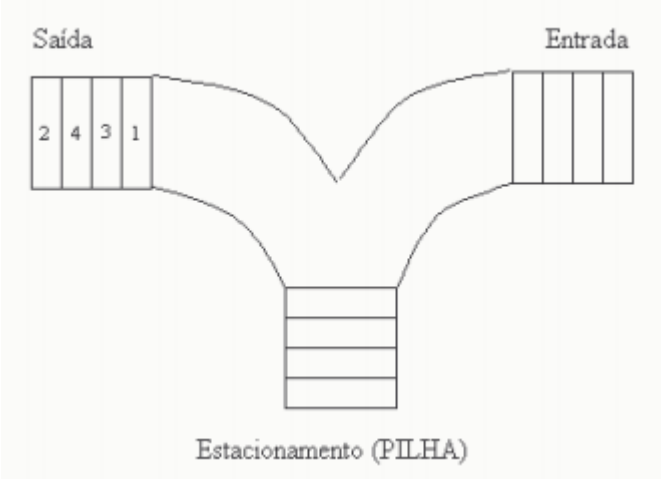
- **Exercício:** resposta

- (b), e após toda a sequência de operações ser efetuada, ter-se-á a seguinte configuração:

(a)



(b)



PUC GOIÁS

Listas Lineares

- **Exercício 2 :** Se existirem seis vagões na Entrada (1 2 3 4 5 6), existe uma seqüência de operações que aplicada sobre a Entrada fornecerá na Saída a ordenação **3 2 5 6 4 1** dos vagões? Se sim, qual é a seqüência de operações? E uma seqüência que forneça a permutação **1 5 4 6 2 3**? Se sim, qual é a seqüência?

PUC GOIÁS

Referências bibliográficas:

[1]. Pícollo, L. Homero. *Estruturas de Dados*. MSD Ed.. Brasília, 2000.

[2]. Knuth, D.E.*The Art of Computer Programming - Vols I e III*. 2nd Edition. Addison Wesley, 1973.