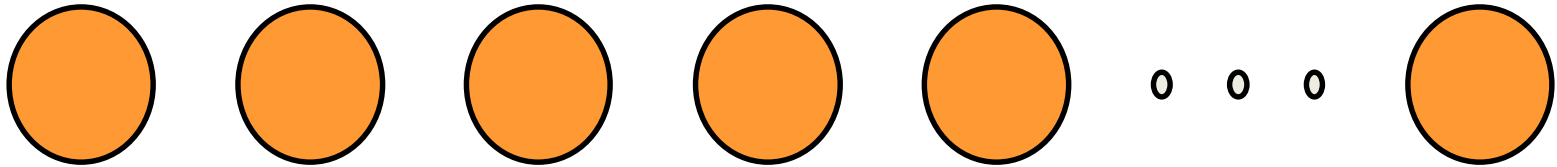


# Listas lineares

Estudo de listas lineares e das operações básicas sobre elas, considerando as diferentes formas de implementação física

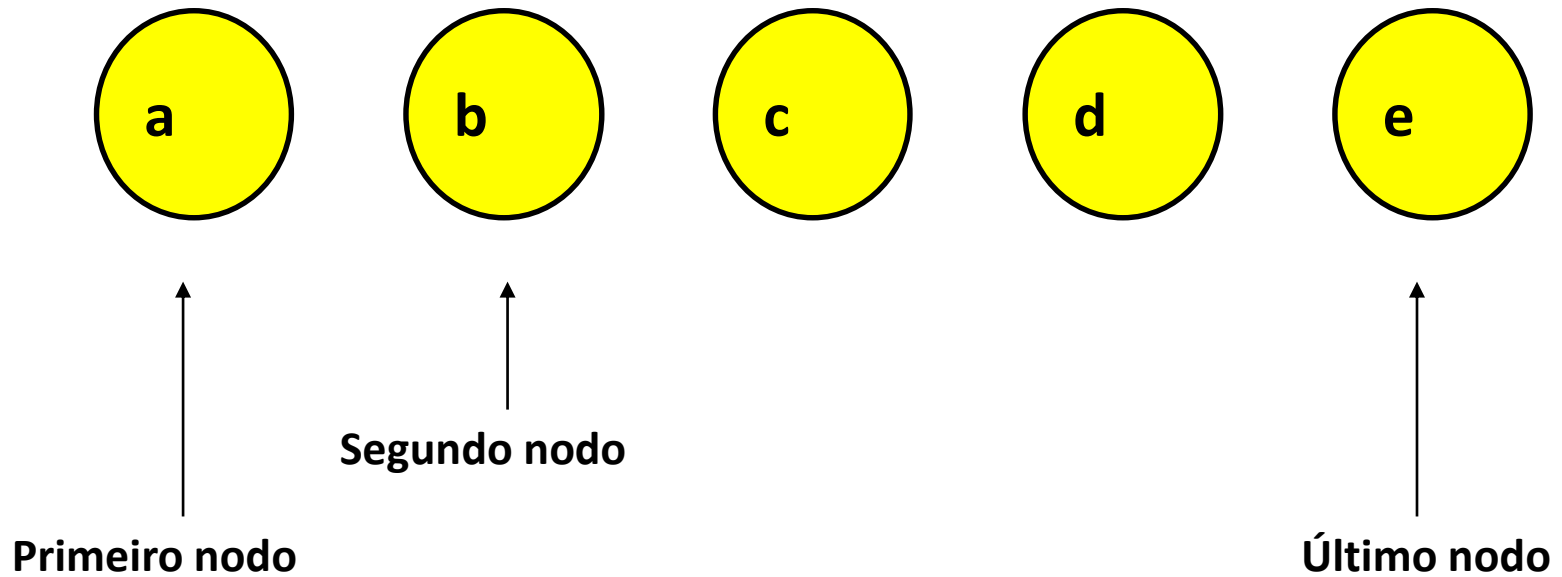
# Lista linear

Uma Lista Linear (LL) é uma **seqüência** de nodos



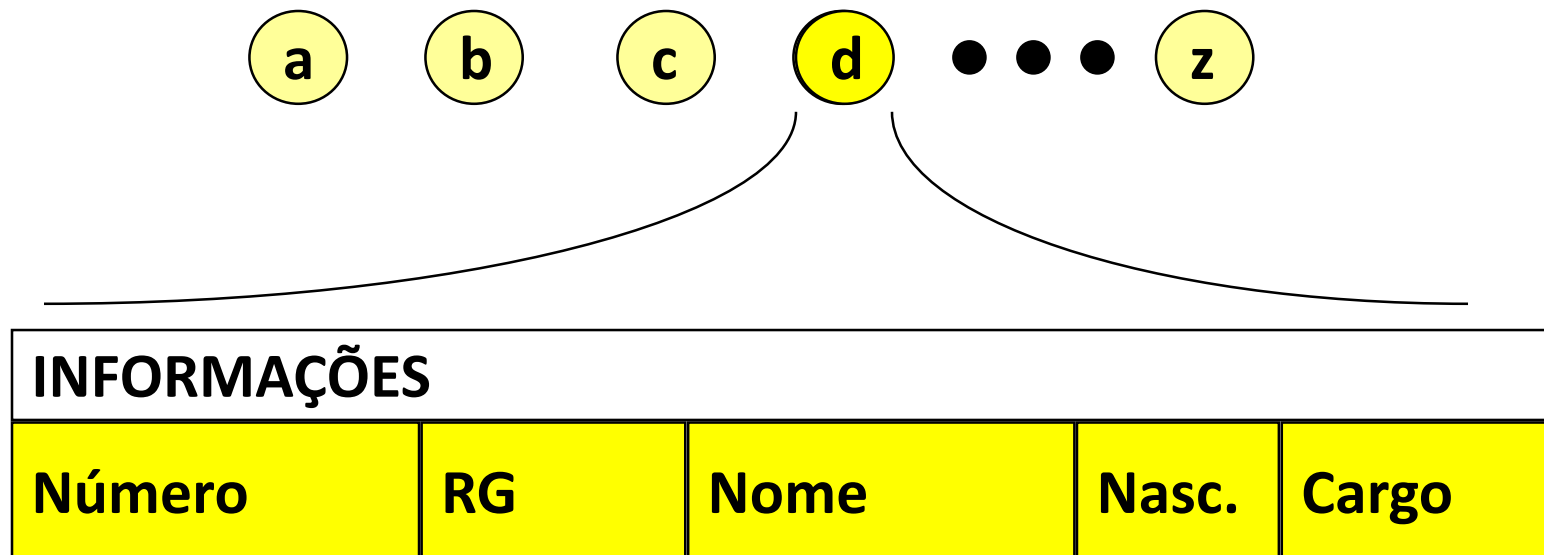
- Nodos - elementos do mesmo tipo
- Relação de ordem linear (ou total)

# Lista linear



# Estrutura dos nodos

- Estrutura interna é abstraída
- Pode ter uma complexidade arbitrária
- Enfatizado o conjunto de relações existente



# Definição formal

Uma lista linear é uma coleção de  $n \geq 0$  nodos  $x_1, x_2, \dots, x_n$ , todos do mesmo tipo, cujas propriedades estruturais relevantes envolvem apenas as posições relativas lineares entre nodos:

$n = 0$  : lista vazia, apresenta zero nodos

$n > 0$  :  $x_1$  é o primeiro nodo

$x_n$  é o último nodo

$1 < k < n$  :  $x_k$  é precedido por  $x_{k-1}$  e sucedido por  $x_{k+1}$

- Lista linear : seqüência de 0 ou mais nodos do mesmo tipo

# Exemplos de aplicações com listas

- Notas de alunos
- Cadastro de funcionários de uma empresa
- Itens em estoque em uma empresa
- Dias da semana
- Letras de uma palavra
- Pessoas esperando ônibus
- Cartas de baralho
- Lista telefonica

# Operações sobre listas lineares

## Operações básicas:

- Criação de uma lista
- Inserção de um nodo
- Exclusão de um nodo
- Acesso a um nodo
- Destruição de uma lista

# listas lineares com disciplina de acesso

- Disciplina de acesso refere-se à forma como os elementos de uma lista linear são acessados, inseridos e removidos.
- Se os elementos de uma lista linear só podem ser inseridos, acessados ou removidos da última posição, chamamos esta lista linear de **pilha (LIFO - Last In First Out)**;



# listas lineares com disciplina de acesso

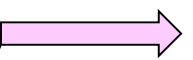
- Se os elementos de uma lista linear só podem ser inseridos na última posição e acessados ou removidos da primeira posição, chamamos esta lista linear de **fila (FIFO - First In First Out)**;

# Considerações sobre alocação de memória

Como armazenar os elementos de uma lista???

A alocação de memória para implementar uma lista pode ser estática ou dinâmica.

# Considerações sobre alocação de memória

- alocação estática
  - área de memória é alocada no momento da compilação
  - Uma lista com alocação estática de memória exige uma definição do número máximo de elementos   
**super** ou **sub** dimensiona-mento do tamanho da lista.

# Considerações sobre alocação de memória

- alocação dinâmica:
  - o espaço de memória é alocado em tempo de execução.
  - Uma lista com alocação dinâmica cresce à medida que novos elementos precisam ser armazenados (e diminui à medida que elementos anteriormente armazenados são retirados da lista).

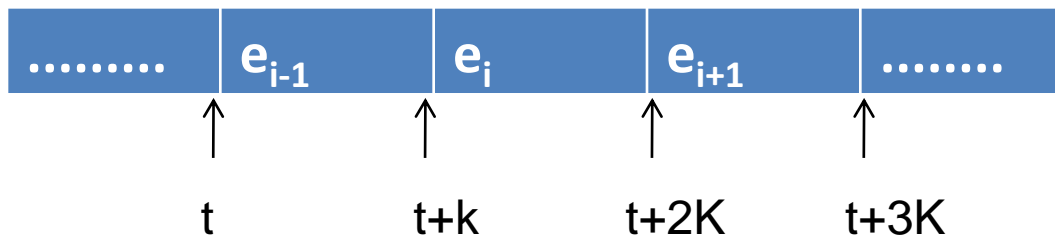
# Considerações sobre o acesso aos elementos de uma lista

- acesso sequencial

--- os elementos de uma lista são armazenados de forma consecutiva na memória.

**Exemplo:** considere que cada elemento da lista tenha tamanho  $k$

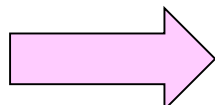
--- o endereço de um elemento  $e_i$  é facilmente calculado



# Considerações sobre o acesso aos elementos de uma lista

- acesso encadeado

--- os elementos de uma lista podem ocupar quaisquer áreas de memória, não necessariamente consecutivas

 para preservar a relação de ordem de uma lista linear, cada elemento da lista deve armazenar sua informação e o endereço de memória onde se encontra o próximo elemento

--- o endereço do elemento  $e_i$  não pode ser facilmente calculado.

# Considerações sobre o acesso aos elementos de uma lista

combinações possíveis:

- alocação estática *versus* alocação dinâmica
- acesso sequencial *versus* acesso encadeado

alocação estática/acesso sequencial	alocação estática/acesso encadeado
alocação dinâmica/acesso sequencial	alocação dinâmica/acesso encadeado

# Referências

- Pereira, S.L. Estruturas de Dados Fundamentais - Conceitos e Aplicações. Editora Érica, 5a. edição, 2001.
- Nina Edelwais e Renata Galante. Série de Livros Didáticos – Informática da UFRGS.