

Lógica de Programação



```
def filter(s: Set, p: Int => Boolean): Set = intersect(s, q)  
def forall(s: Set, p: Int => Boolean): Boolean = {  
  def iter(a: Int): Boolean = {  
    if (contains(p,a) && !contains(s,a)) false  
    else if (a > 1000) true  
    else iter(a + 1)  
  }  
  iter(-1000)  
}
```

```
def exists(s: Set, p: Int => Boolean): Boolean = {  
  (x: Int) => !(forall(s, !p(x)))  
}  
  
def map(s: Set, f: Int => Int): Set = {  
  (b: Int) => exists(s, (c: Int) => f(c) == b)  
}
```

Estruturas de Dados

- São formas de armazenar e organizar dados na memória do computador;
- Tipos de Dados Abstratos (ADT):
 - O que importa é **o que** a estrutura faz e não **como** ela faz.
- Principais tipos de estruturas de dados ADT:
 - **Lista**
 - **Pilha**
 - **Fila**
 - **Registro**

Lista

Características

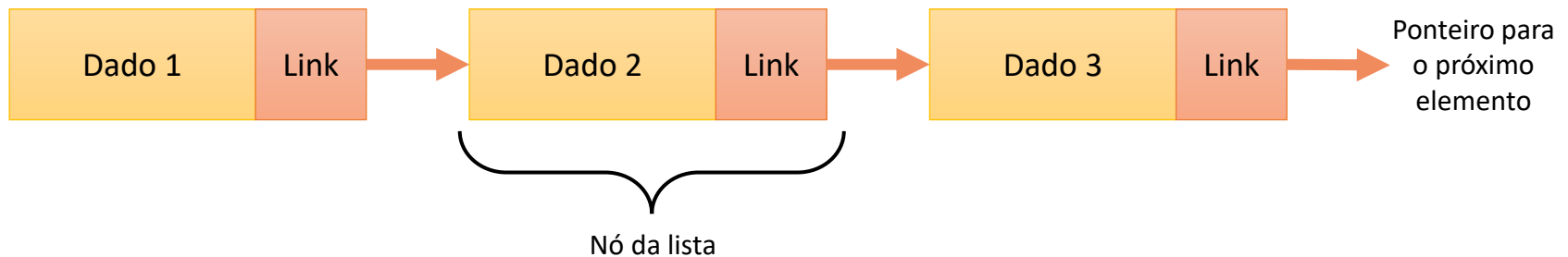
- Os dados são armazenados de forma sequencial, em estruturas chamadas de NÓ;
- Principal tipo é a Lista Encadeada (Linked List);
- Os NÓS são formados por um campo de informação e um campo endereço (ponteiro, link) que conecta ao próximo NÓ da lista;
- Ao contrário dos Vetores que são estáticos, as Listas tem tamanho Dinâmico;
- Pode ser Simples ou Dupla e também pode ser do tipo Circular.



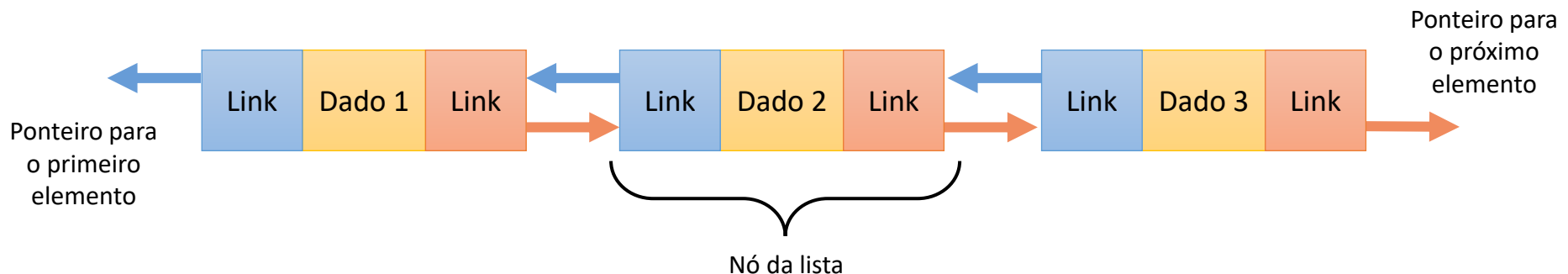
Lista

Arranjo na Memória

■ Lista Encadeada



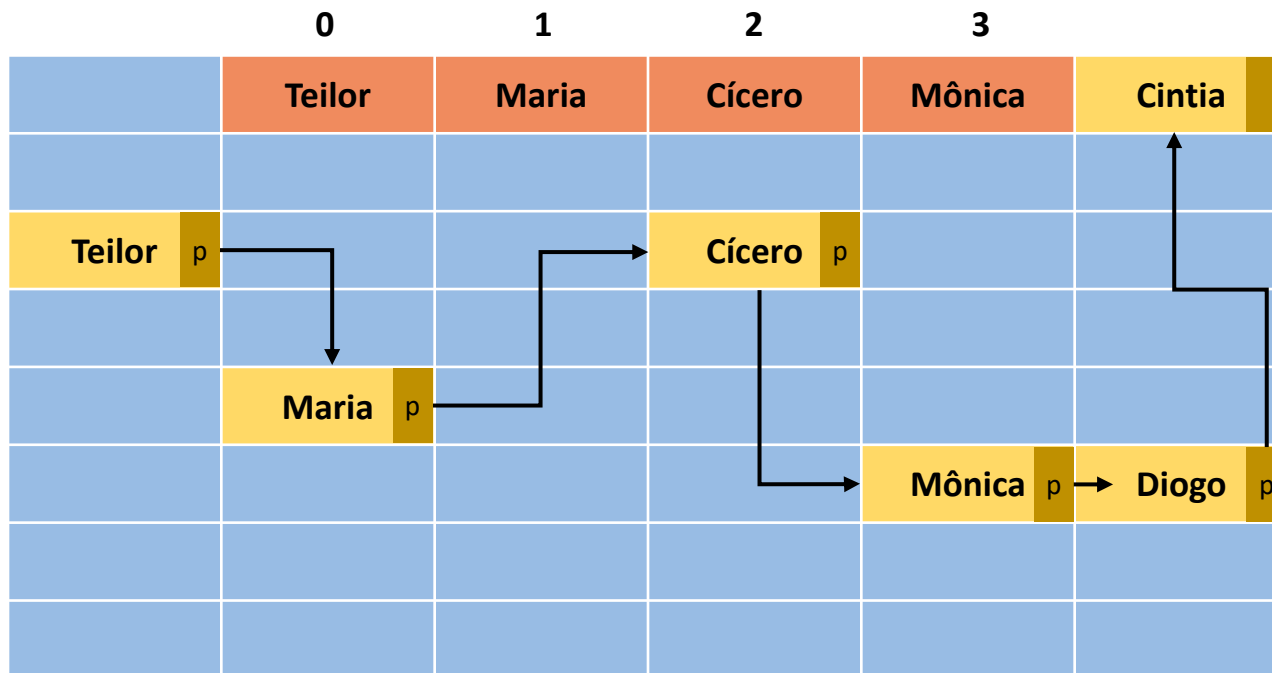
■ Lista Duplamente Encadeada



Lista

Lista x Vetor

- Memória do Computador:



Lista

Operações em Listas

- **pegar() / get()** : Retornar um elemento específico da lista;
- **inserir() / insert()** : Inserir um elemento em qualquer posição da lista;
- **remover() / remove()** : Remover um elemento da lista;
- **substituir() / replace()** : Substituir um elemento específico da lista;
- **tamanho() / size()** : Verificar o tamanho da lista (quantidade de elementos);
- **estaVazio() / isEmpty()** : Verificar se a lista está vazia;
- **estaCheio() / isFull()** : Verificar se a lista está cheia.

Lista

Exemplos de Uso

- Lista de convidados de um casamento;
- Lista de pedidos de uma lanchonete;
- Lista de funcionários contratados;
- Lista de alunos aprovados.

Pilha

Características

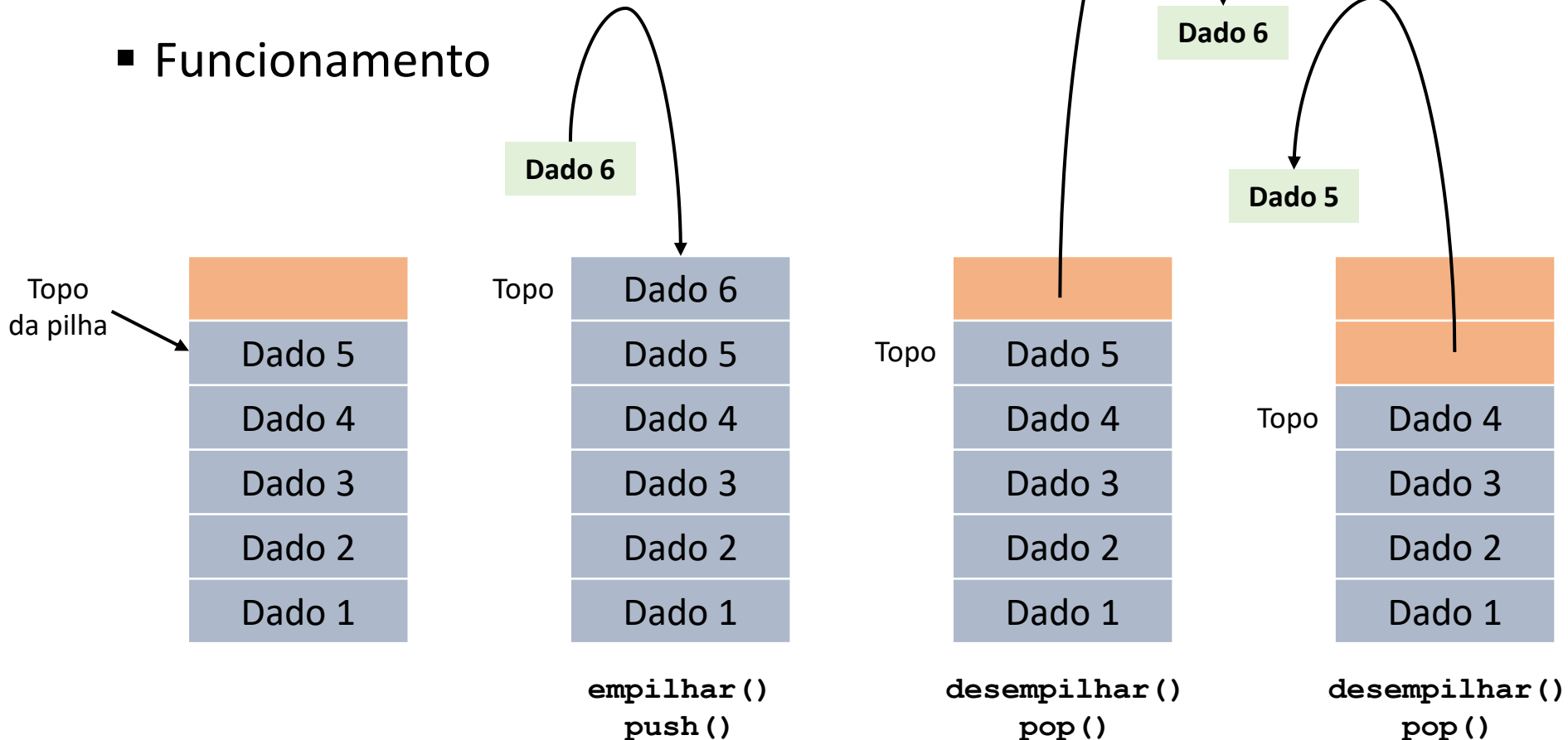
- Dados armazenados de forma linear;
- Os dados são inseridos e removidos de acordo com o princípio LIFO (last-in-first-out);
- Os elementos podem ser inseridos e removidos apenas de um lado da estrutura, chamado de TOPO;
- Usada quando precisamos acessar o elemento mais recente adicionado.



Pilha

Arranjo na Memória

■ Funcionamento



Pilha

Operações em Pilhas

- **empilhar() / push()** : Inserir um item no topo da pilha;
- **desempilhar() / pop()** : Remover o elemento do topo da pilha;
- **imprimir() / peek()** : Visualizar o item do topo da pilha, sem remove-lo;
- **tamanho() / size()** : Verificar o tamanho da pilha (nº de itens);
- **estaVazio() / isEmpty()** : Verificar se a pilha está vazia;
- **estaCheio() / isFull()** : Verificar se a pilha está cheia.

Pilha

Exemplos de Uso

- Mecanismos de Desfazer (Ctrl+Z) e Refazer (Ctrl+R);
- Processamento de Texto;
- Rotas de mapas;
- Implementação de funções recursivas;

Fila

Características

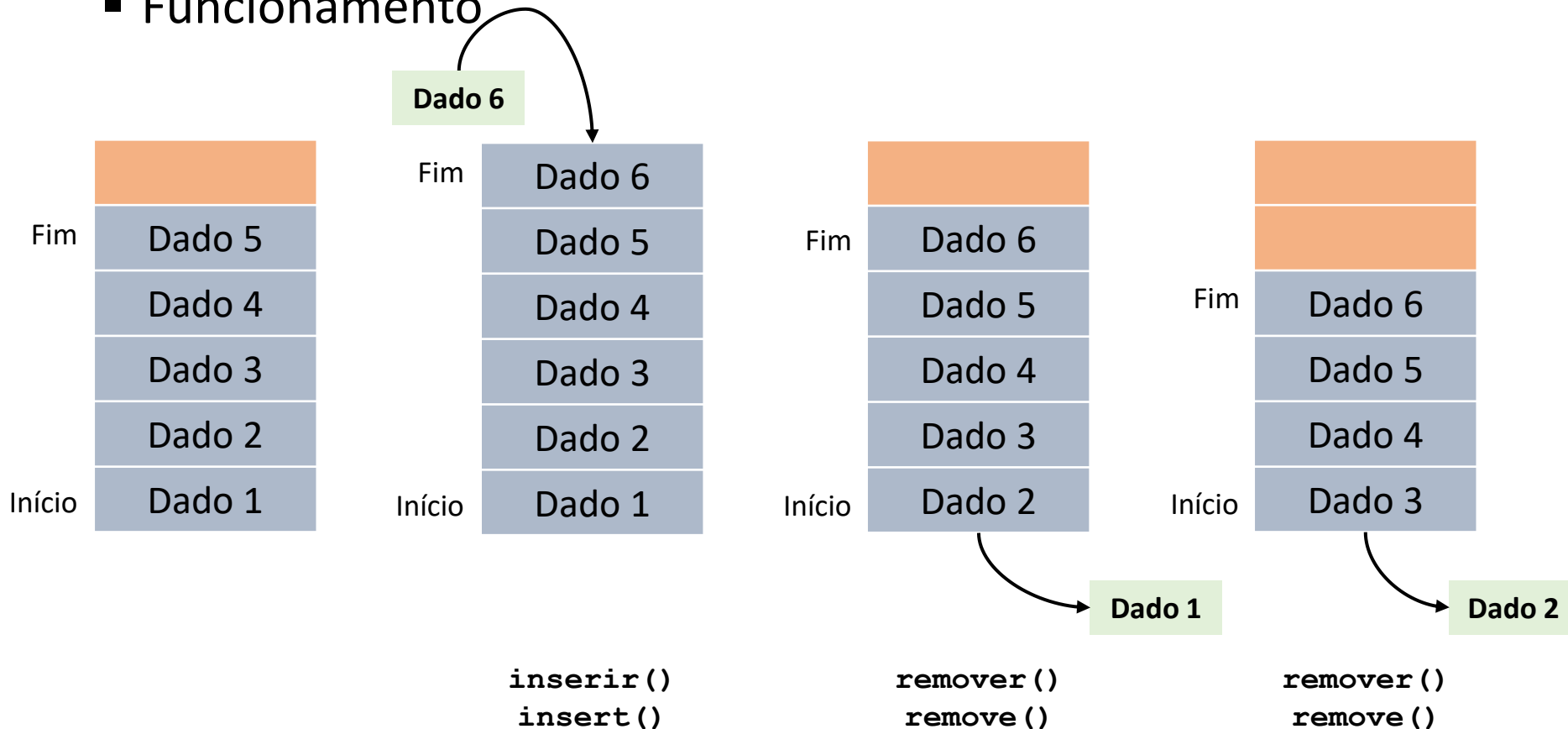
- Dados armazenados de forma linear também;
- Os dados são inseridos e removidos de acordo com o princípio FIFO (first-in-first-out);
- Os elementos são inseridos de um lado da estrutura, o FINAL, e só podem ser removidos do outro lado, o INÍCIO;
- Usada quando precisamos acessar os elementos em ordem de chegada, ou prioridade.



Fila

Arranjo na Memória

■ Funcionamento



Fila

Operações em Filas

- **enfileirar() / enqueue()** : Inserir um item no final da fila;
- **desenfileirar() / dequeue()** : Remover o item do início da fila;
- **imprimir() / peek()** : Visualizar um item específico da fila, sem remove-lo;
- **tamanho() / size()** : Verificar o tamanho da fila (nº de itens);
- **estaVazio() / isEmpty()** : Verificar se a fila está vazia;
- **estaCheio() / isFull()** : Verificar se a fila está cheia.

Fila

Exemplos de Uso

- Respostas para requisições em serviços compartilhados;
- Filas de impressões;
- Acesso a discos;
- Processos em acesso a CPU;
- Transferência de dados (buffer, E/S);
- Gerenciamento de filas por senhas (Ex: Hospital, Banco e etc.);
- Listas de espera em geral.

Registro

Características

- ...