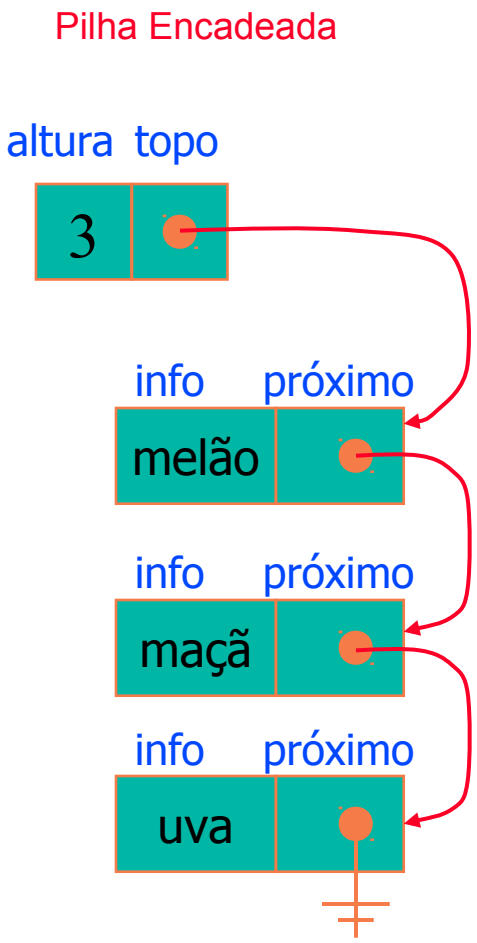
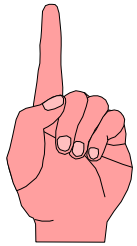
Filas Encadeadas

Extensões do Conceito de Lista Encadeada

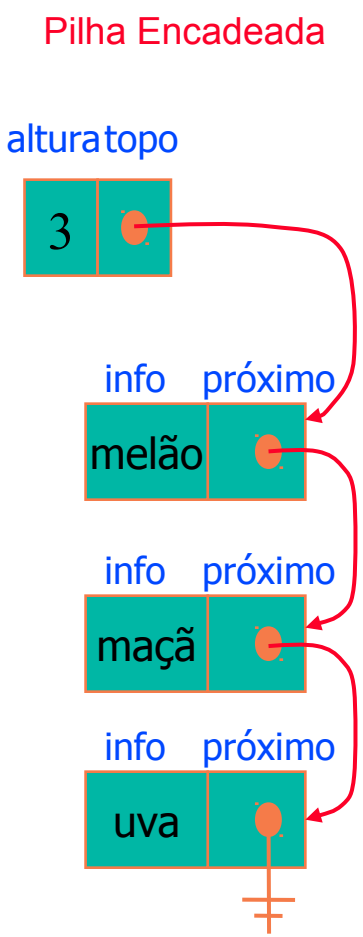
- A idéia da Lista Encadeada como foi vista até agora é o modelo mais geral e simples.
- Pode ser especializada e extendida das mais variadas formas:
- Especializada:
  - Pilhas encadeadas
  - Filas
- Extendida:
  - Listas Duplamente Encadeadas
  - Listas Circulares Simples e Duplas



Pilhas Encadeadas

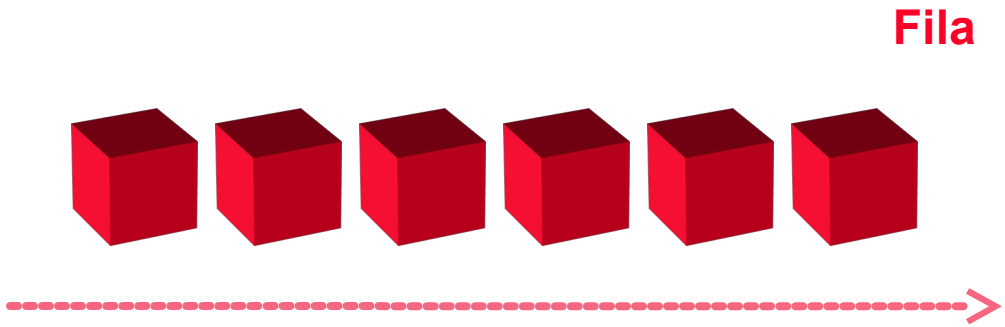


- São pilhas onde os elementos são encadeados.
  - Não há estouro de pilha.
  - Como o acesso de dados na pilha é puramente seqüencial, não há perda de eficiência.
- Somente duas operações existem:
  - Empilhar: equivale a adicionar no início
  - Desempilhar: equivale a retirar do início.
- É a forma como tradicionalmente pilhas são implementadas.



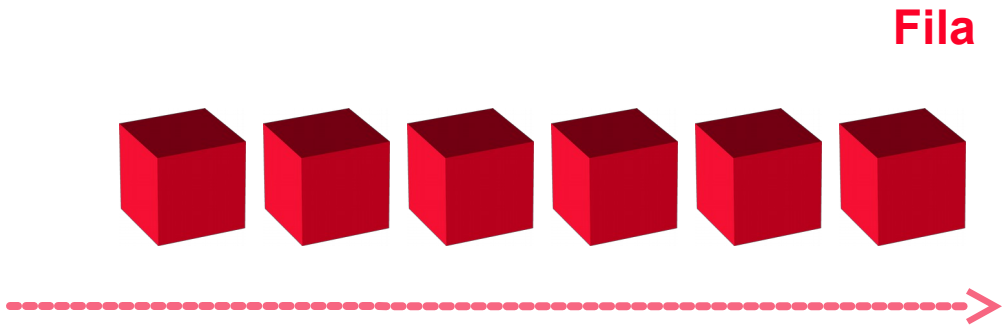
Filas

- A Fila é uma estrutura de dados que simula uma fila da vida real.
- Possui duas operações básicas:
  - Incluir no fim da fila
  - Retirar do começo da fila.
  - Chamada de Estrutura-FIFO:
    First-In, First-Out - O primeiro que entrou é o primeiro a sair.



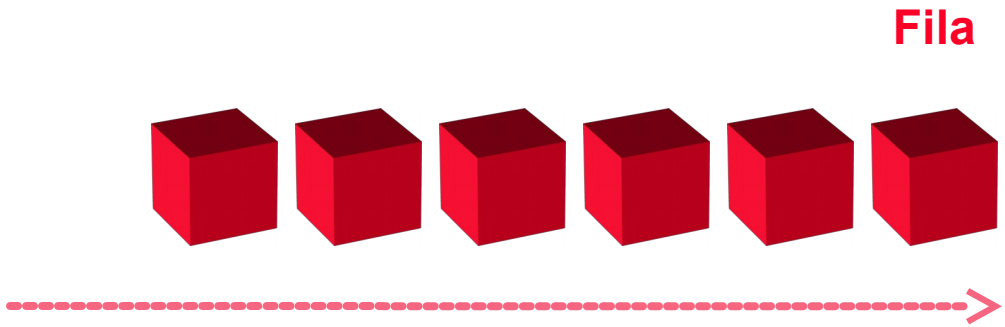
Filas

- É uma estrutura de dados improtantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - Chão de fábrica: fila de camisetas a serem estampadas.
    - Comércio: simulação de fluxo de um caixa de supermercado.
    - Tráfego: simulação de um cruzamento com um semáforo.



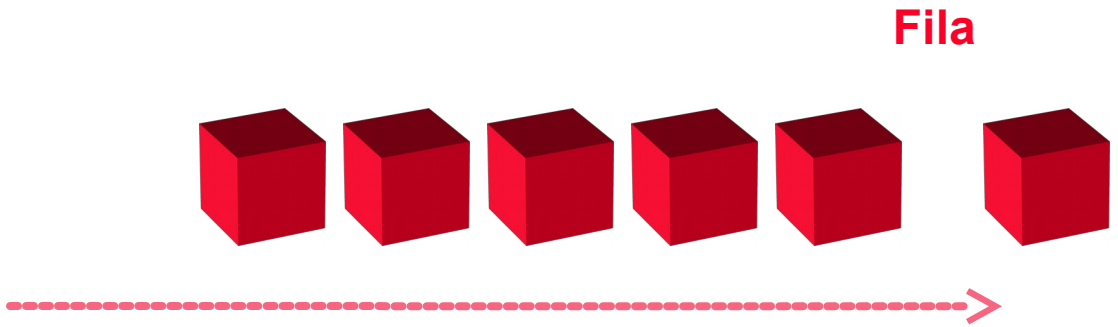
Filas

- É uma estrutura de dados improtantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - Chão de fábrica: fila de camisetas a serem estampadas.
    - Comércio: simulação de fluxo de um caixa de supermercado.
    - Tráfego: simulação de um cruzamento com um semáforo.



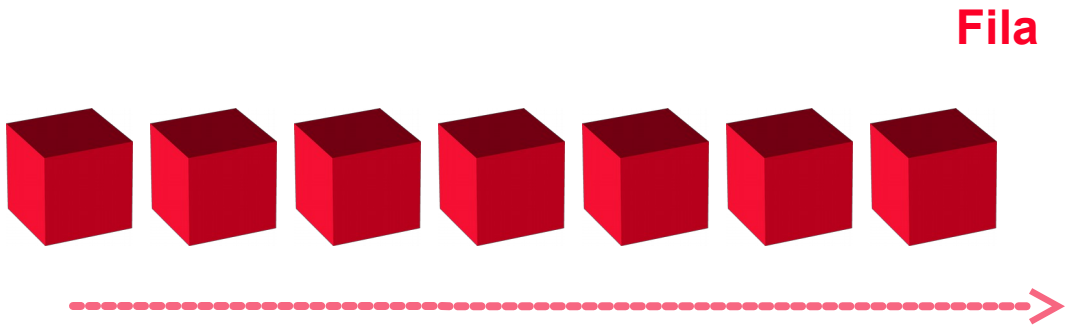
Filas

- É uma estrutura de dados improtantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - Chão de fábrica: fila de camisetas a serem estampadas.
    - Comércio: simulação de fluxo de um caixa de supermercado.
    - Tráfego: simulação de um cruzamento com um semáforo.



Filas

- É uma estrutura de dados improtantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - Chão de fábrica: fila de camisetas a serem estampadas.
    - Comércio: simulação de fluxo de um caixa de supermercado.
    - Tráfego: simulação de um cruzamento com um semáforo.

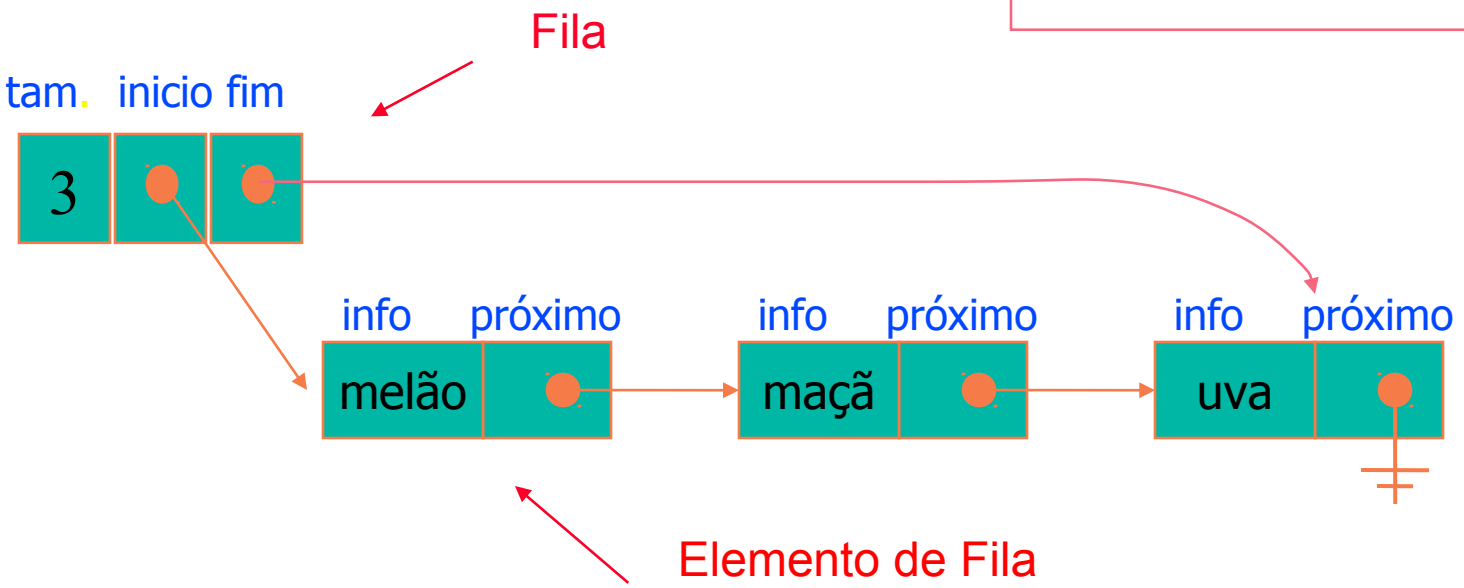


Filas - Representação

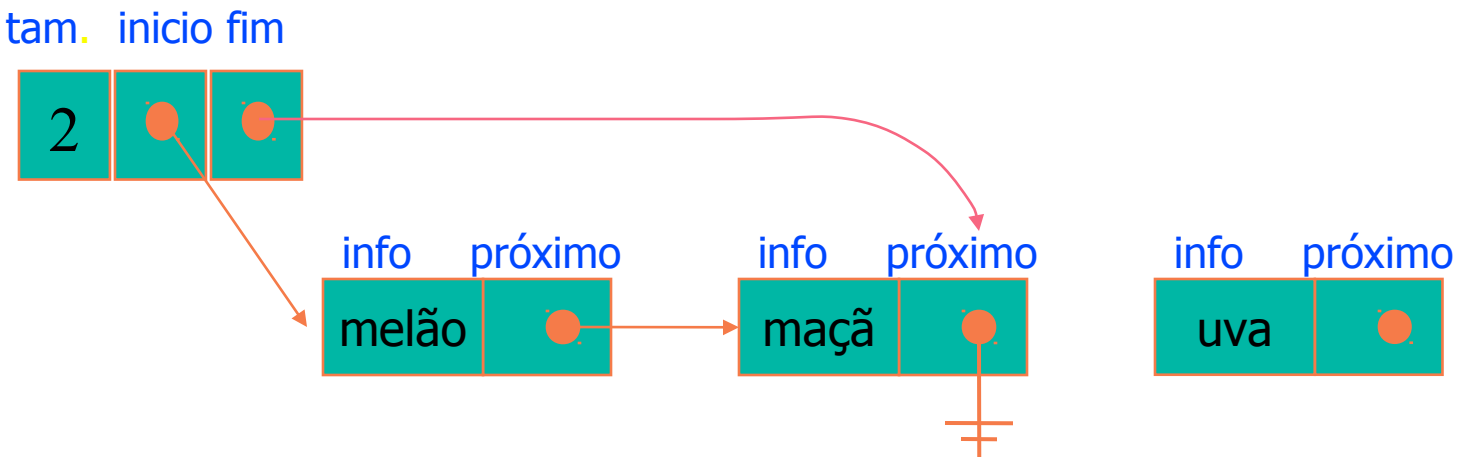
- Extensão da lista encadeada
  - Referenciamos o último elemento também.
  - Adicionamos no fim
  - Excluímos do início

Pseudo-código:

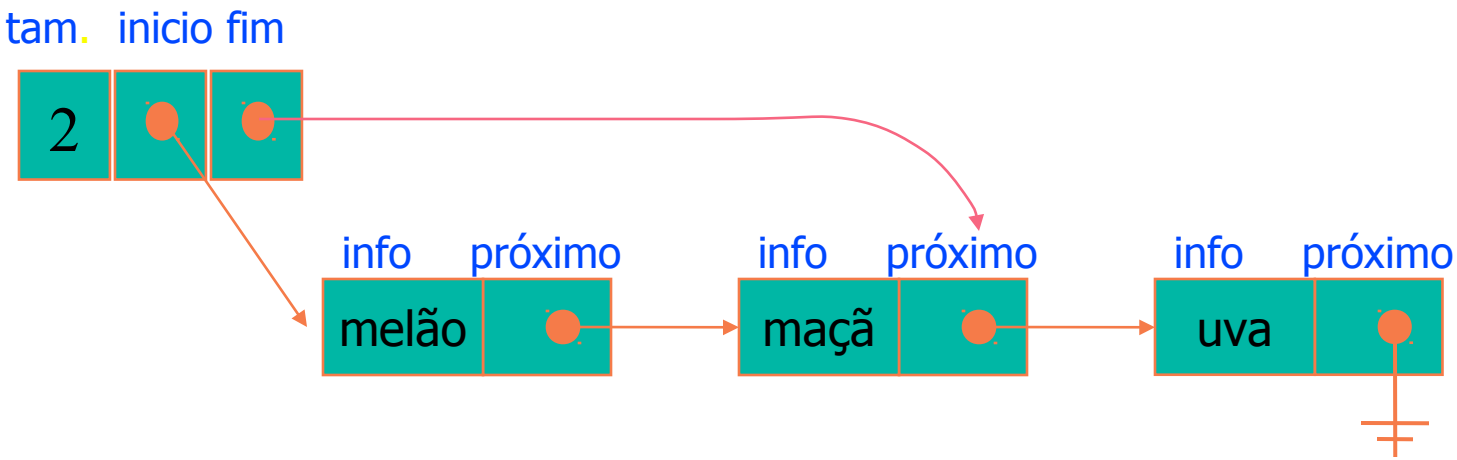
```
tipo Fila {
   Elemento *inicio;
   Elemento *fim;
   inteiro  tamanho;
};
```



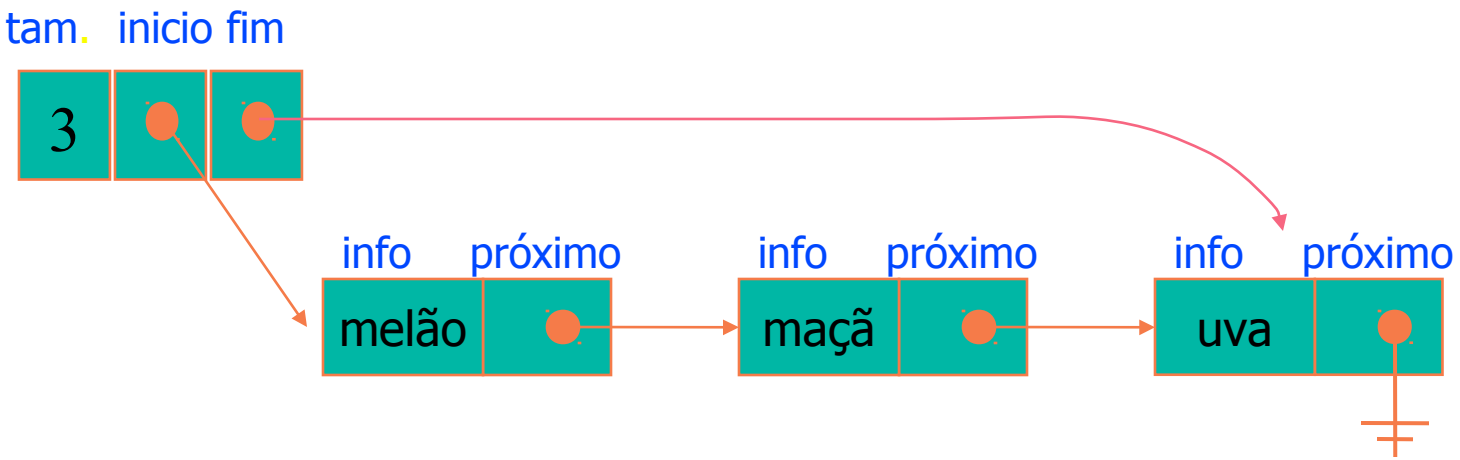
Algoritmo Adiciona Fila



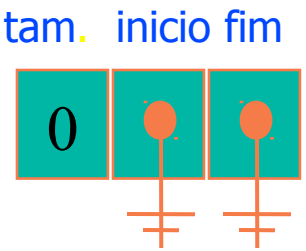
Algoritmo Adiciona Fila

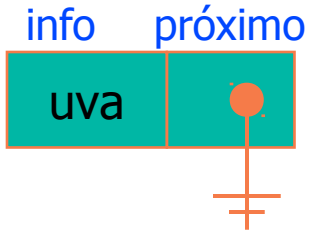


Algoritmo Adiciona Fila

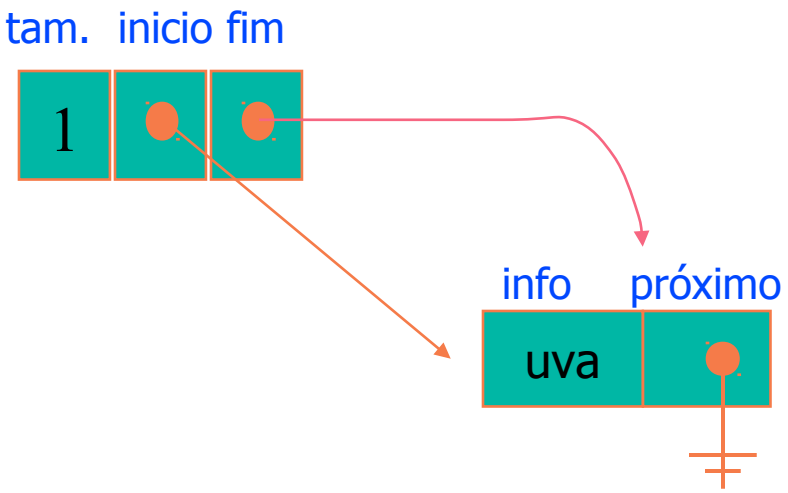


Algoritmo Adiciona Fila – fila vazia

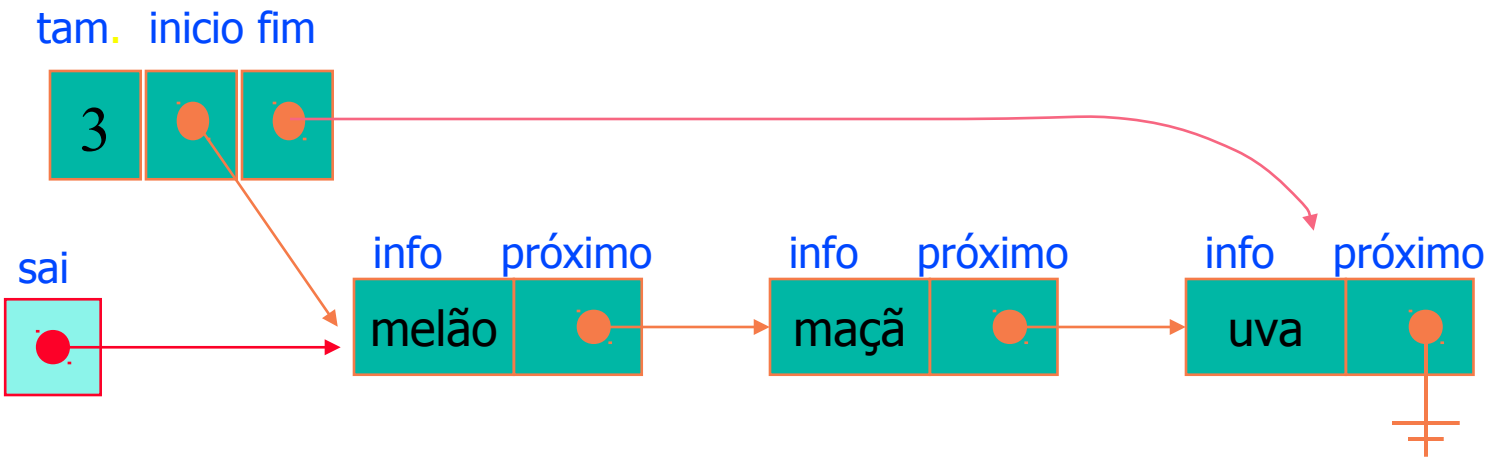




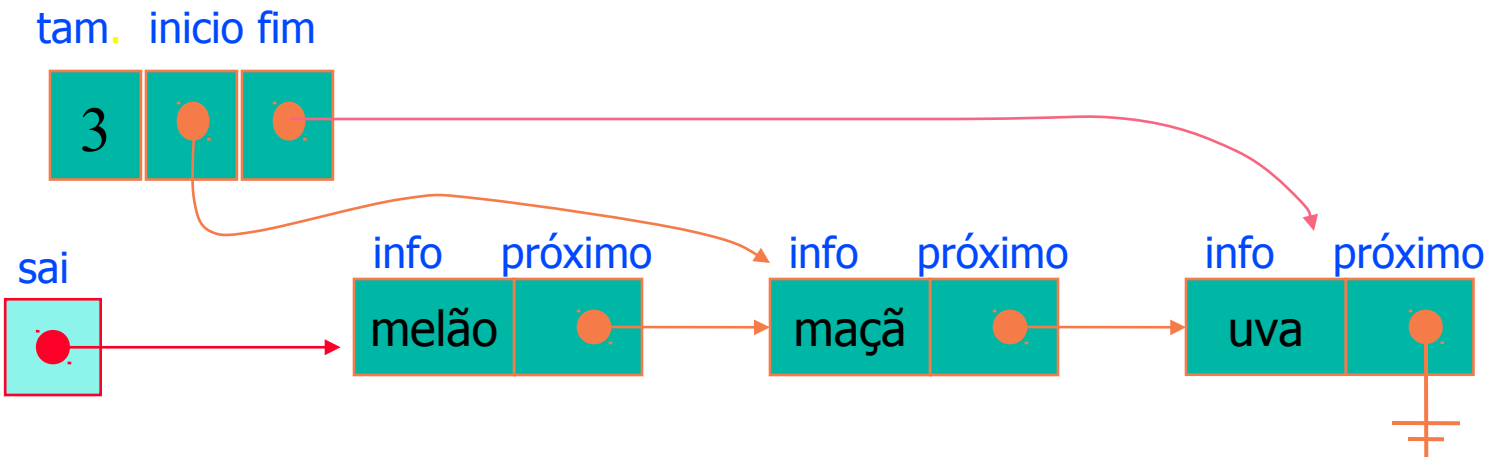
Algoritmo Adiciona Fila – fila vazia



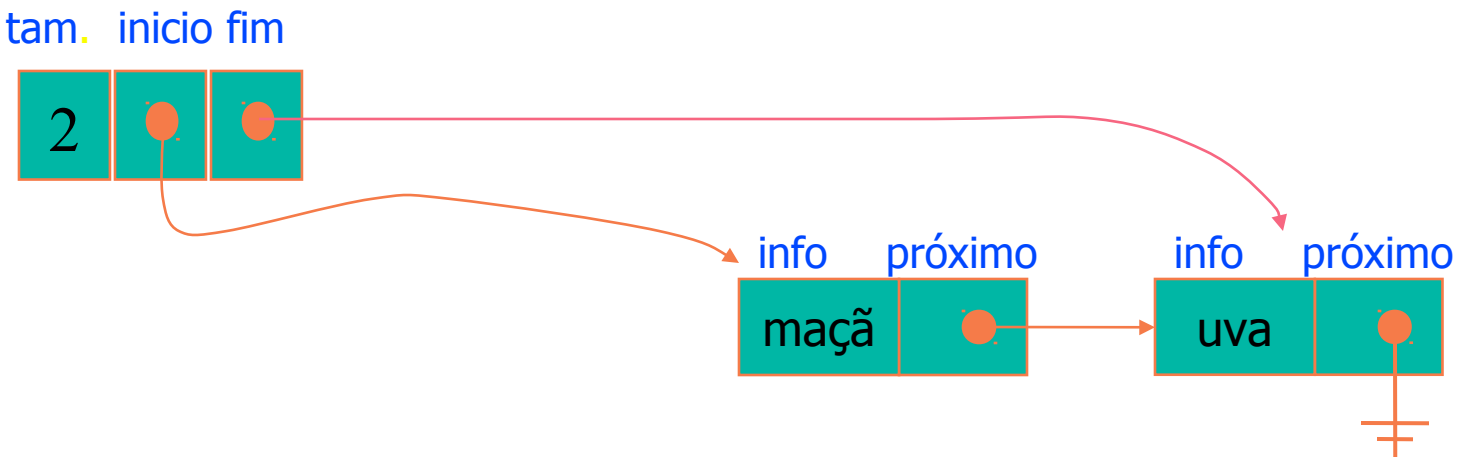
Algoritmo Retira Fila



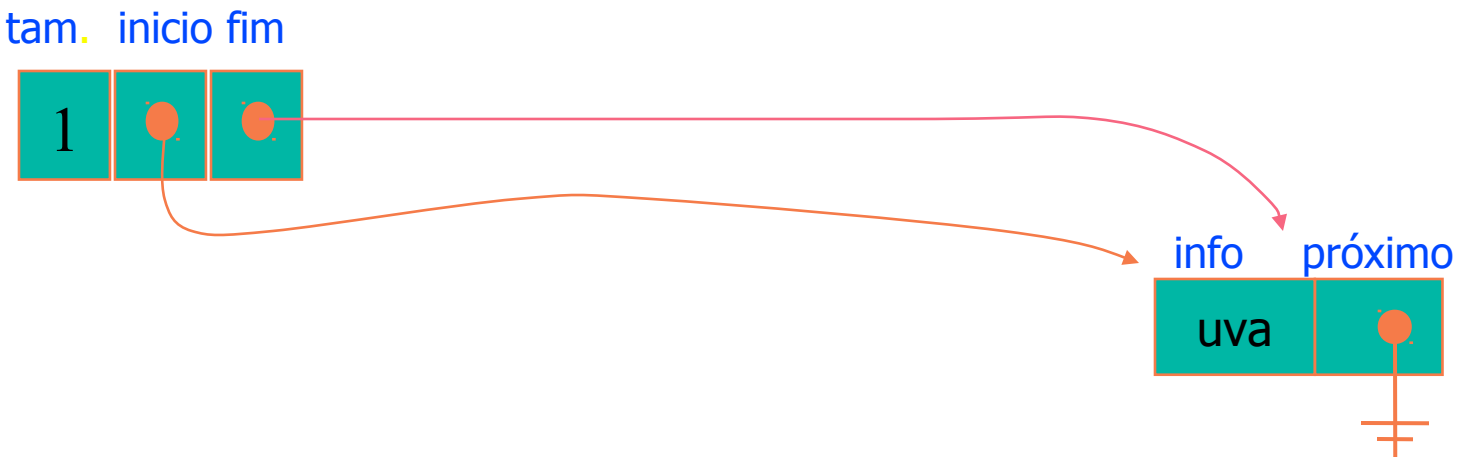
Algoritmo Retira Fila



Algoritmo Retira Fila

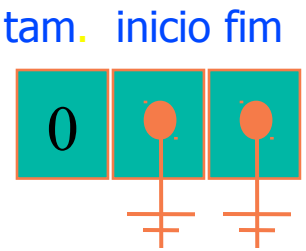


Algoritmo Retira Fila – fila unitária



Não preciso de uma variável auxiliar sai

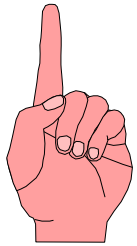
Algoritmo Retira Fila – fila unitária



Exercício

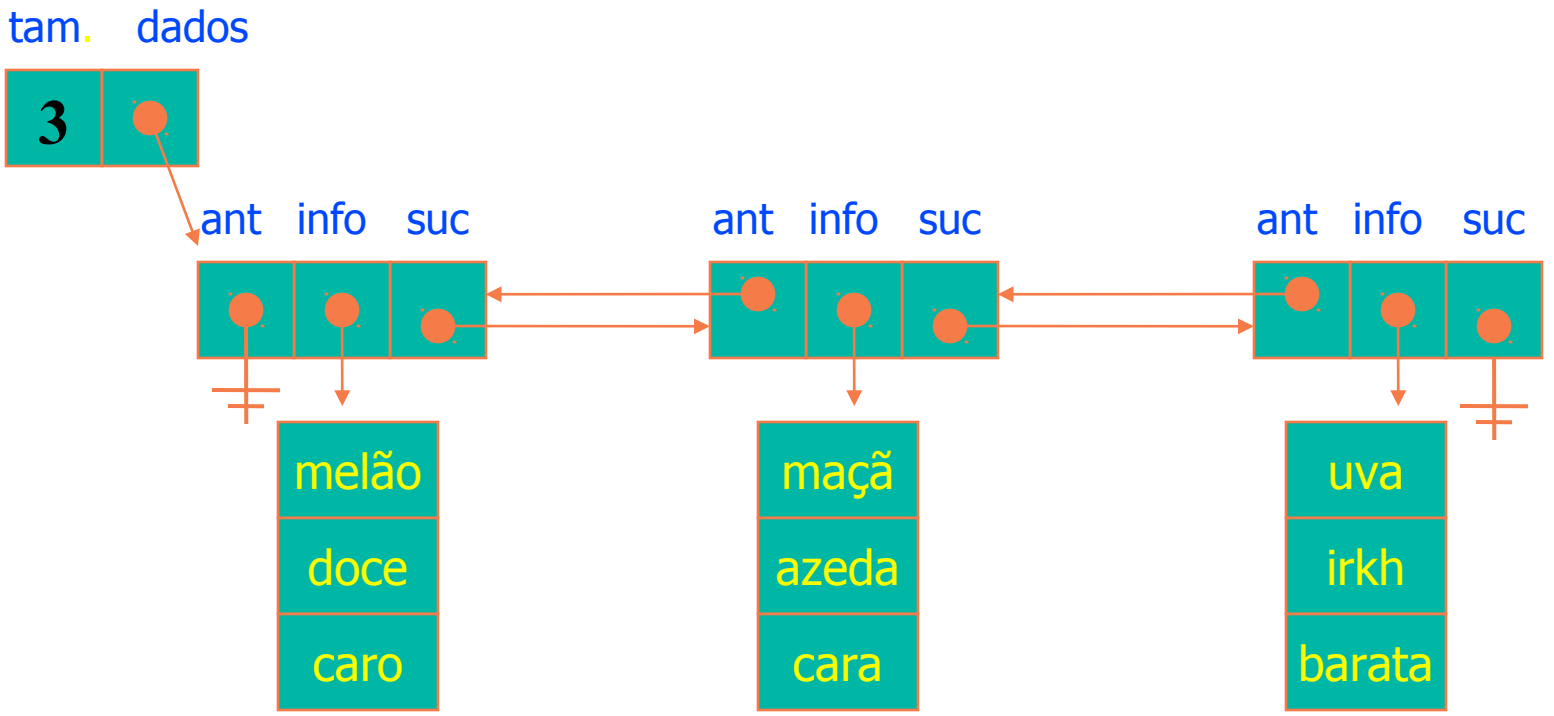
- Implemente o TAD Fila com as suas duas operações de manipulação.
- Implemente um programa principal de aplicação que utilize uma fila para:
  - Ler dados de pedidos de solicitação de conserto de computadores da marca HAL pela sua assistencia técnica AssistHAL.
  - Utilize o seguinte TipoInfo: Nome do Solicitante, Telefone, Data da Entrega (inserida automaticamente), Modelo do Computador e Valor do Conserto, que é um valor chutado.
- Após cadastrado um pedido, o sistema deverá informar quando o solicitante poderá contar com o computador pronto, sendo que:
  - A oficina AssistHAL leva em média duas semanas por pedido.

Lista Duplamente Encadeada



- A Lista Encadeada e a Fila possuem a desvantagem de somente podermos caminhar em uma direção.
  - Vimos que para olhar um elemento que “acabamos de passar” precisamos de uma variável auxiliar “anterior”.
  - Para olhar outros elementos ainda anteriores não temos nenhum meio, a não ser começar de novo.
- A Lista Duplamente Encadeada é uma estrutura de lista que permite deslocamento em ambos os sentidos:
  - Útil para representar conjuntos de eventos ou objetos a serem percorridos em dois sentidos.
  - Ex.: Itinerários de ônibus, trem ou avião.
  - Útil também quando realizamos uma busca aproximada e nos movemos para e frente e trás.

Lista Duplamente Encadeada : modelagem



Modelagem Cabeça de Lista Dupla

- Necessitamos:
  - Um ponteiro para o primeiro elemento da lista.
  - Um inteiro para indicar quantos elementos a lista possue.
- Pseudo-código:

```
tipo ListaDupla {
  ElementoDuplo  *dados;
  inteiro        tamanho;
};
```

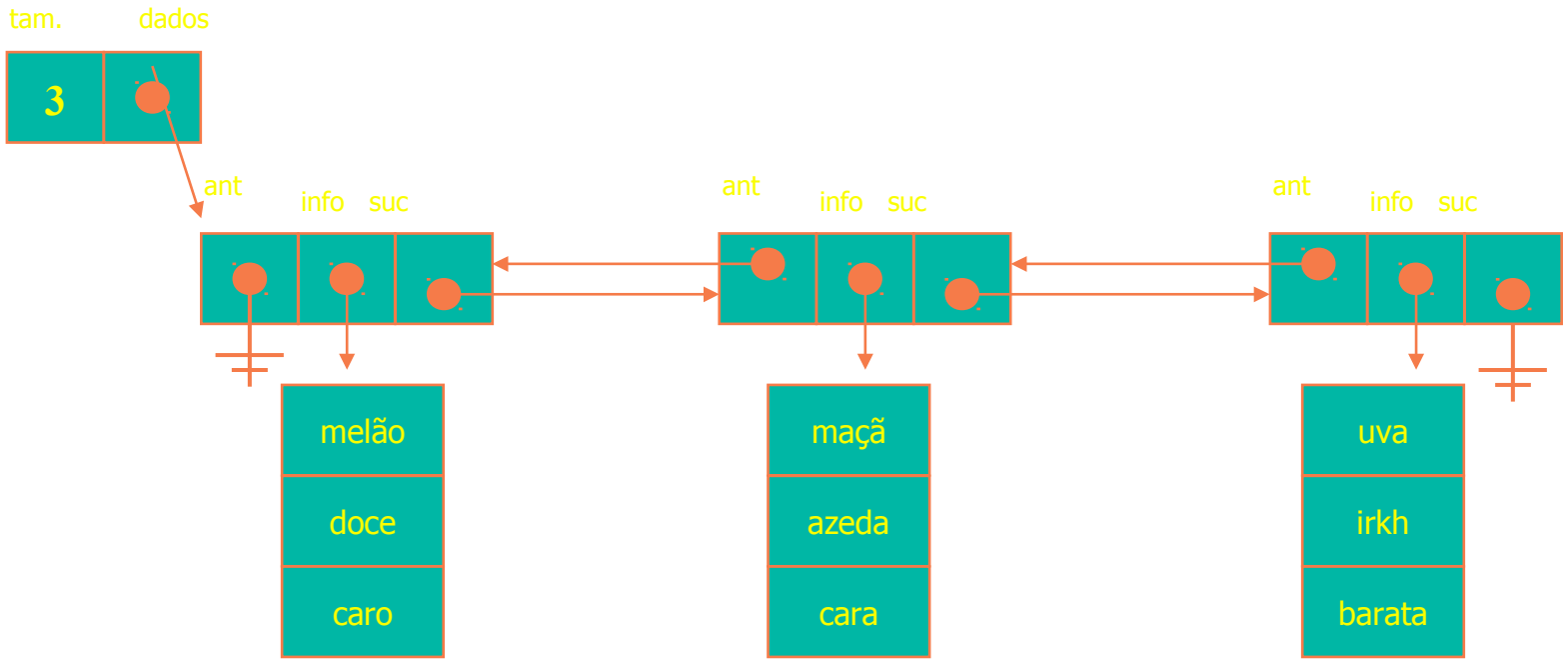
Modelagem Cabeça de Lista Dupla

- Necessitamos:
  - Um ponteiro para o elemento anterior na lista.
  - Um ponteiro para o elemento sucessor na lista.
  - Um ponteiro para a informação que vamos armazenar.
- Pseudo-código:

```
tipo ElementoDuplo {
  ElementoDuplo  *anterior;
  ElementoDuplo  *sucessor;
  TipoInfo       *info;
};
```

Modelagem Cabeça de Lista Dupla

- Aspecto Funcional:
  - Colocar e retirar dados da lista.
  - Testar se a lista está vazia e outros testes.
  - Inicializa-la e garantir a ordem dos elementos.



Modelagem de Lista Duplamente Encadeada

- Operações: Colocar e retirar dados da lista:
  - AdicionaDuplo(listaDupla, dado)
  - AdicionaNoInícioDuplo(listaDupla, dado)
  - AdicionaNaPosiçãoDuplo(listaDupla, dado, posição)
  - AdicionaEmOrdemDuplo(listaDupla, dado)

  - RetiraDuplo(listaDupla)
  - RetiraDoInícioDuplo(listaDupla)
  - RetiraDaPosiçãoDuplo(listaDupla, posição)
  - RetiraEspecíficoDuplo(listaDupla, dado)

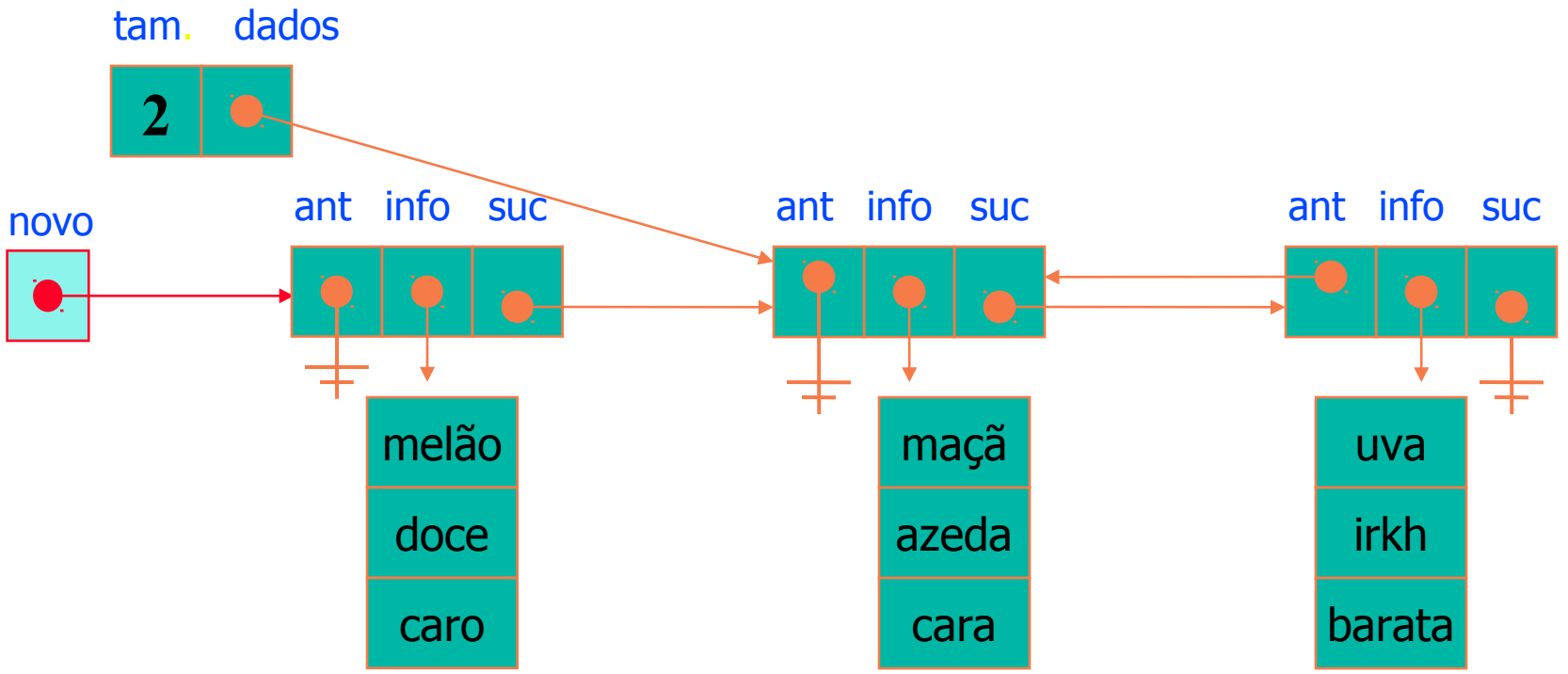
Modelagem de Lista Duplamente Encadeada

- Operações: Testar a lista e outros testes:
  - ListaVaziaDuplo(listaDupla)
  - PosicaoDuplo(listaDupla, dado)
  - ContemDuplo(listaDupla, dado)
- Operações: Inicializar ou limpar:
  - CriaListaDupla()
  - DestroiListaDupla(listaDupla)

Algoritmo AdicionarNoIncioListaDupla

- Procedimento:
  - Testamos se é possível alocar um elemento.
  - Fazemos o sucessor deste novo elemento ser o primeiro da lista.
  - Fazemos o seu antecessor ser NULO.
  - Fazemos a cabeça de lista apontar para o novo.
- Parâmetros:
  - O tipo info (dado) a ser inserido.
  - ListaDupla.

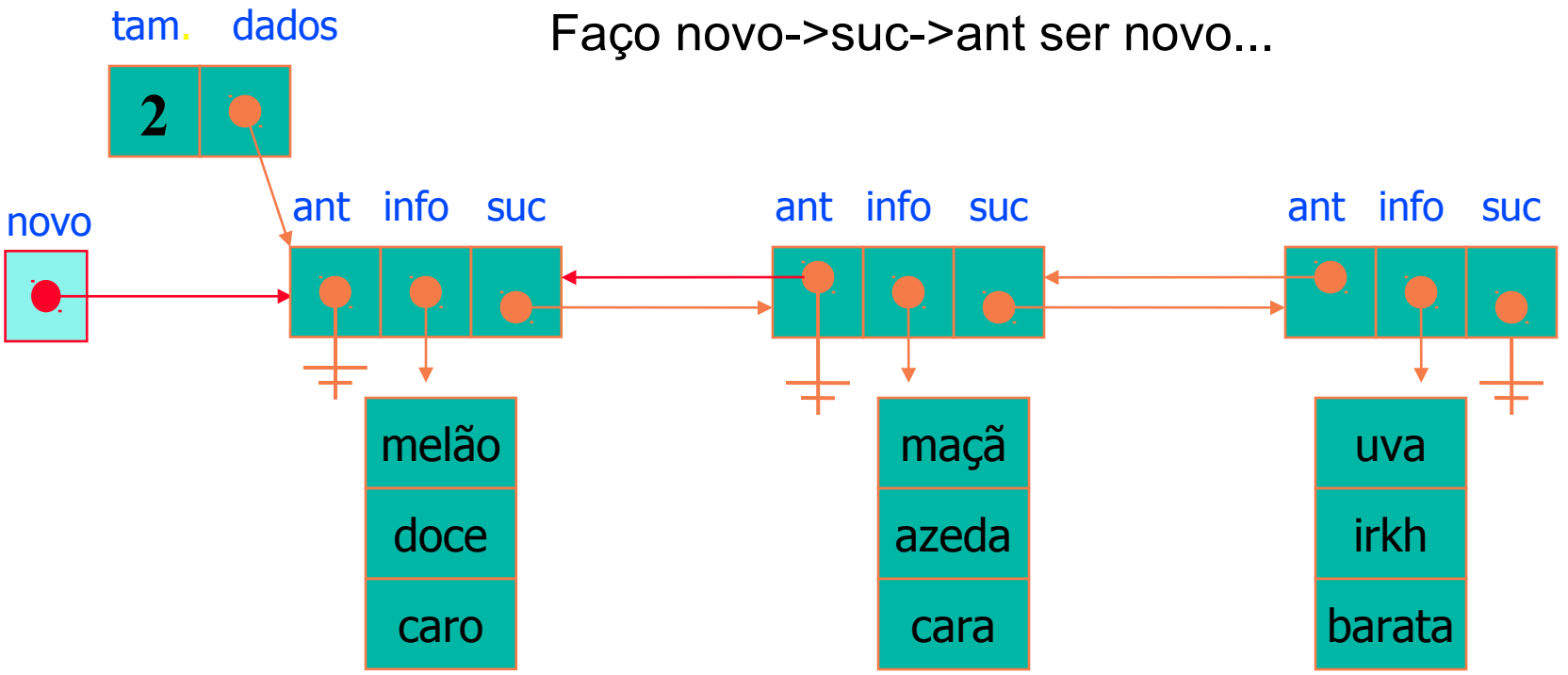
Algoritmo AdicionarNoIncioListaDupla



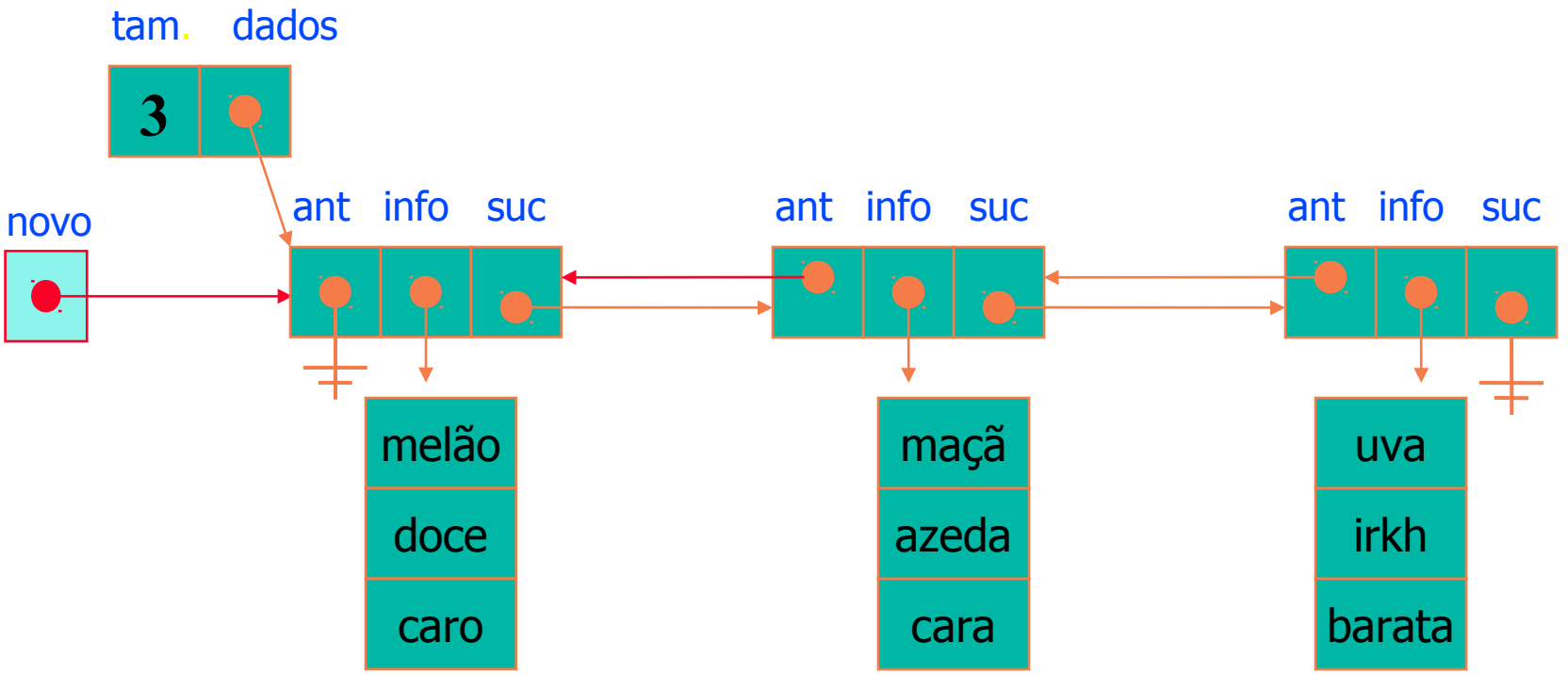
Algoritmo AdicionarNoIncioListaDupla

Caso novo->suc não seja nulo...

Faço novo->suc->ant ser novo...



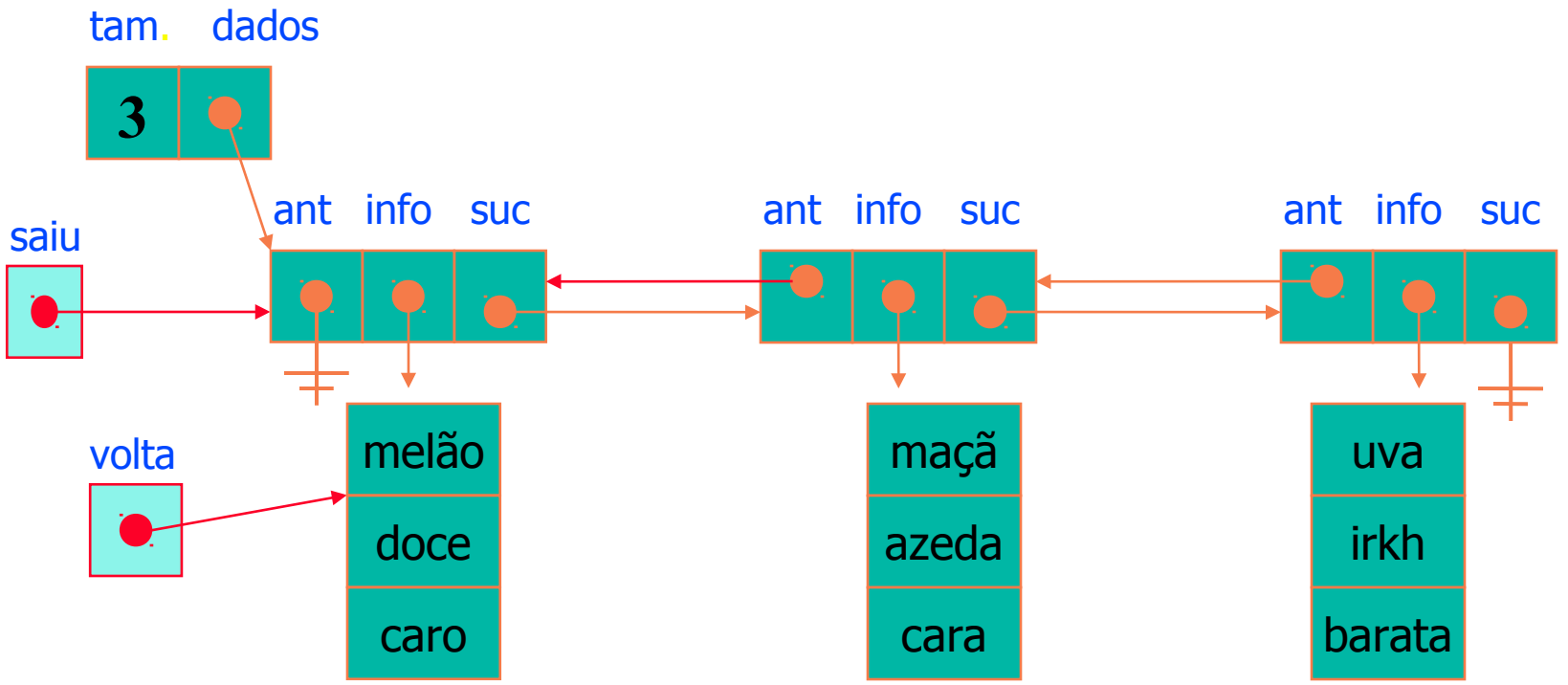
Algoritmo AdicionarNoIncioListaDupla



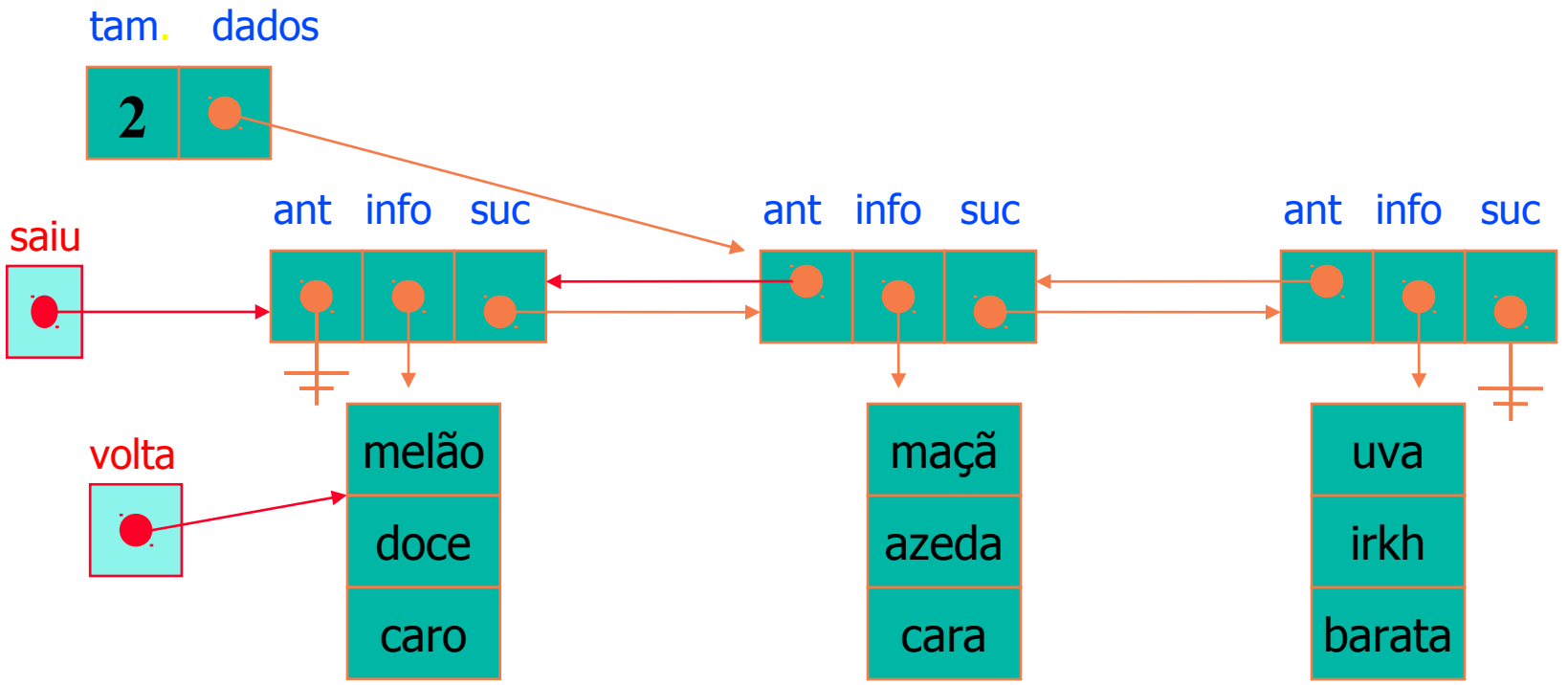
Algoritmo RetirarInicioListaDupla

- Procedimento:
  - Testamos se há elementos.
  - Decrementamos o tamanho.
  - Se o elemento possuir sucessor, o antecessor do sucessor será NULO.
  - Liberamos a memória do elemento.
  - Devolvemos a Informação.
- Parâmetros:
  - ListaDupla.

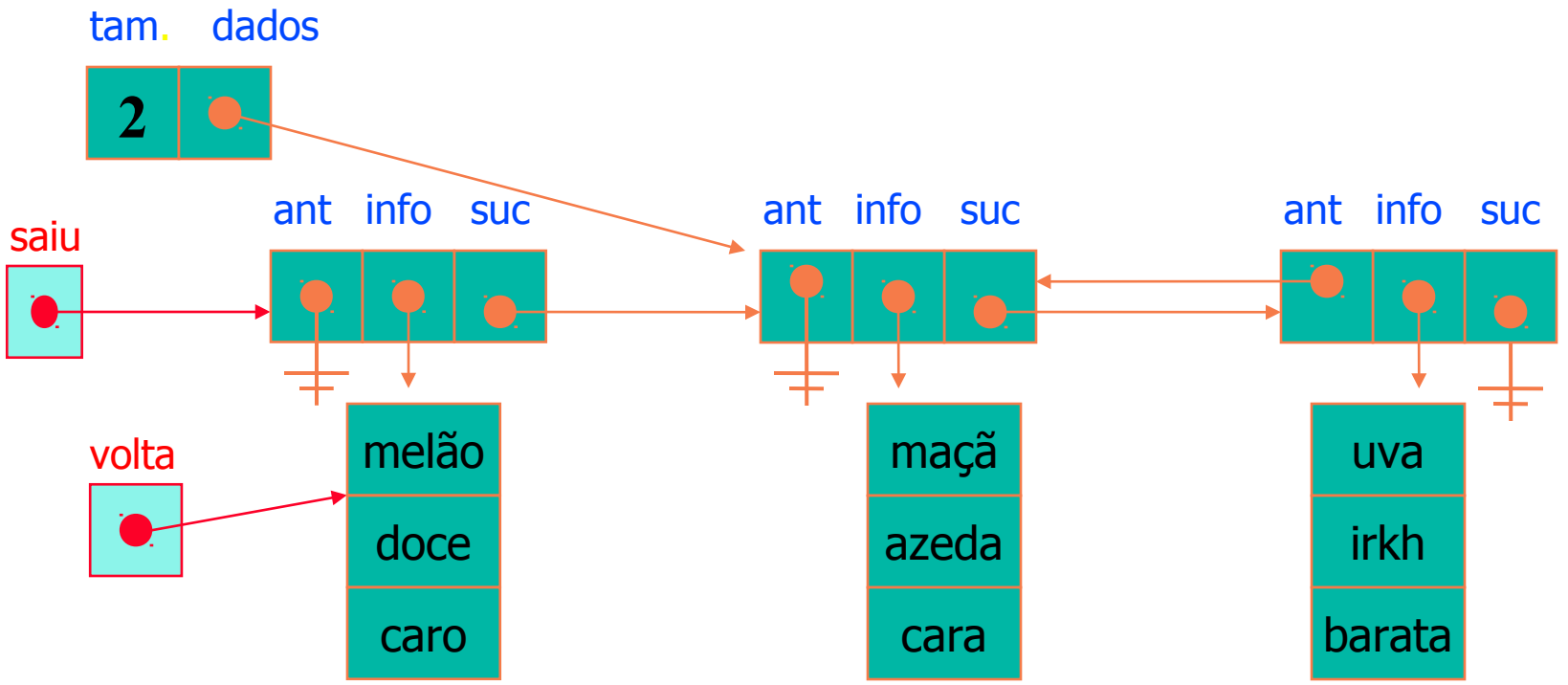
Algoritmo RetirarInicioListaDupla



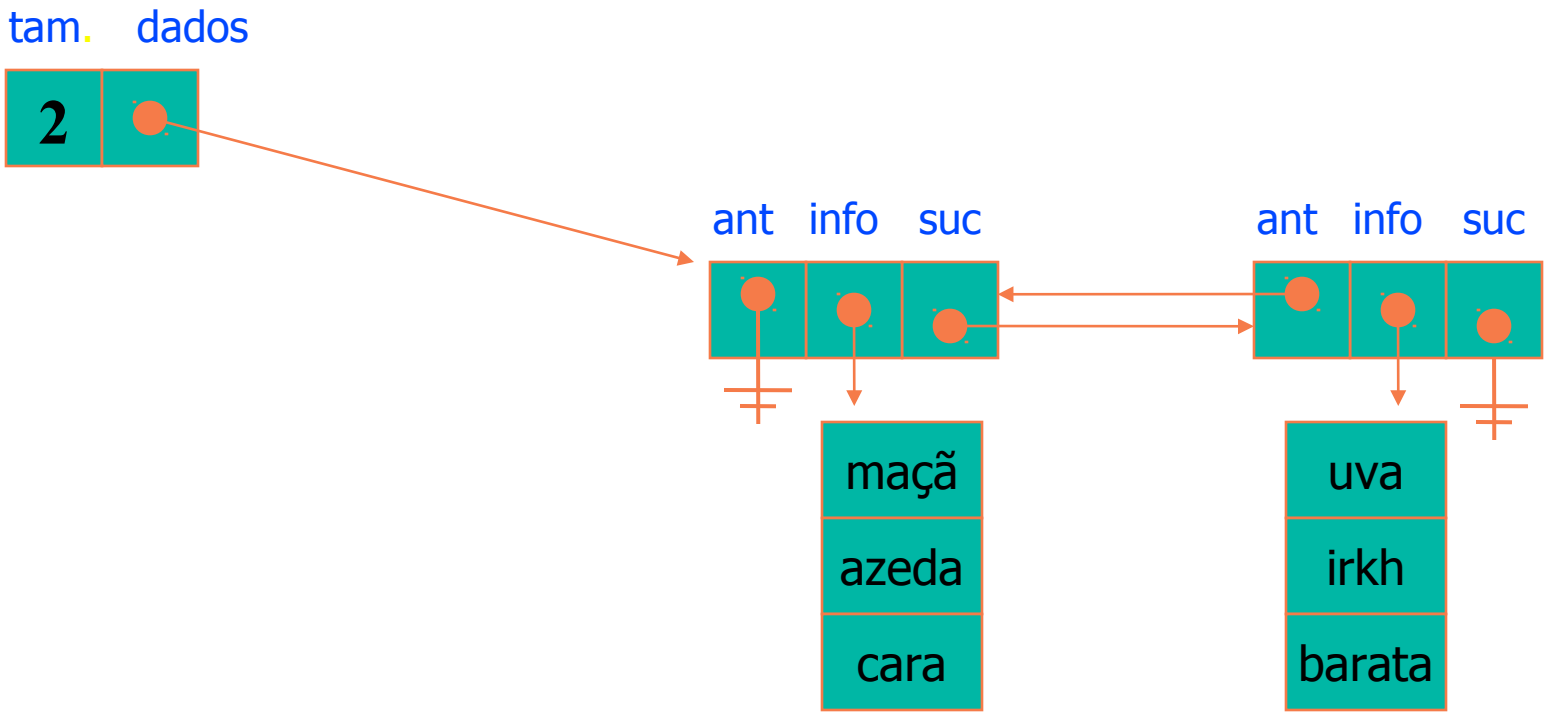
Algoritmo RetirarInicioListaDupla



Algoritmo RetirarInicioListaDupla



Algoritmo RetirarInicioListaDupla



Algoritmo AdicionaPosiçãoListaDupla

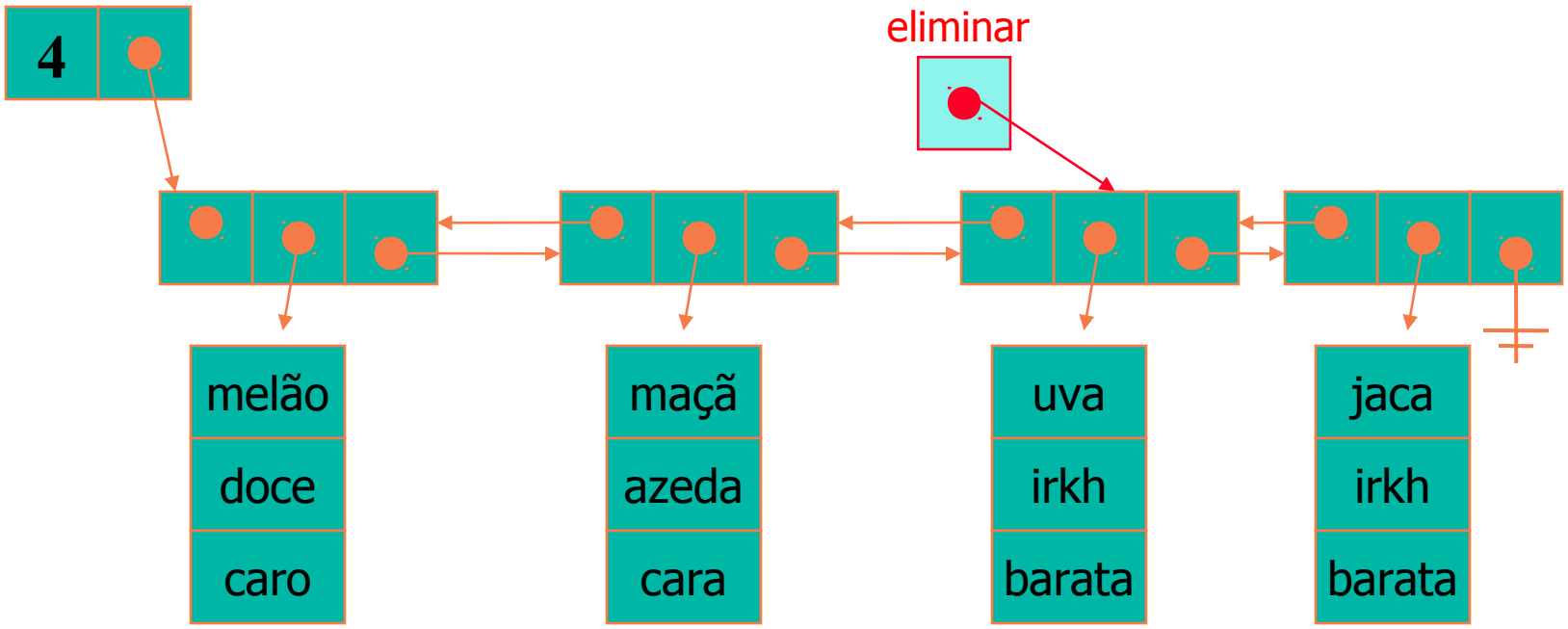
- Praticamente idêntico à lista encadeada.
- Procedimento:
  - Testamos se a posição existe e se é possível alocar elemento.
  - Caminhamos até a posição.
  - Adicionamos o novo dado na posição.
  - Incrementamos o tamanho.
- Parâmetros:
  - O dado a ser inserido.
  - A posição onde inserir.
  - Lista.

Algoritmo AdicionaPosiçãoListaDupla

- Mais simples que a Lista Encadeada.
- Procedimento:
  - Testamos se a posição existe.
  - Caminhamos até a posição.
  - Retiramos o dado da posição.
  - Decrementamos o tamanho.
- Parâmetros:
  - A posição de onde retirar.
  - Lista.

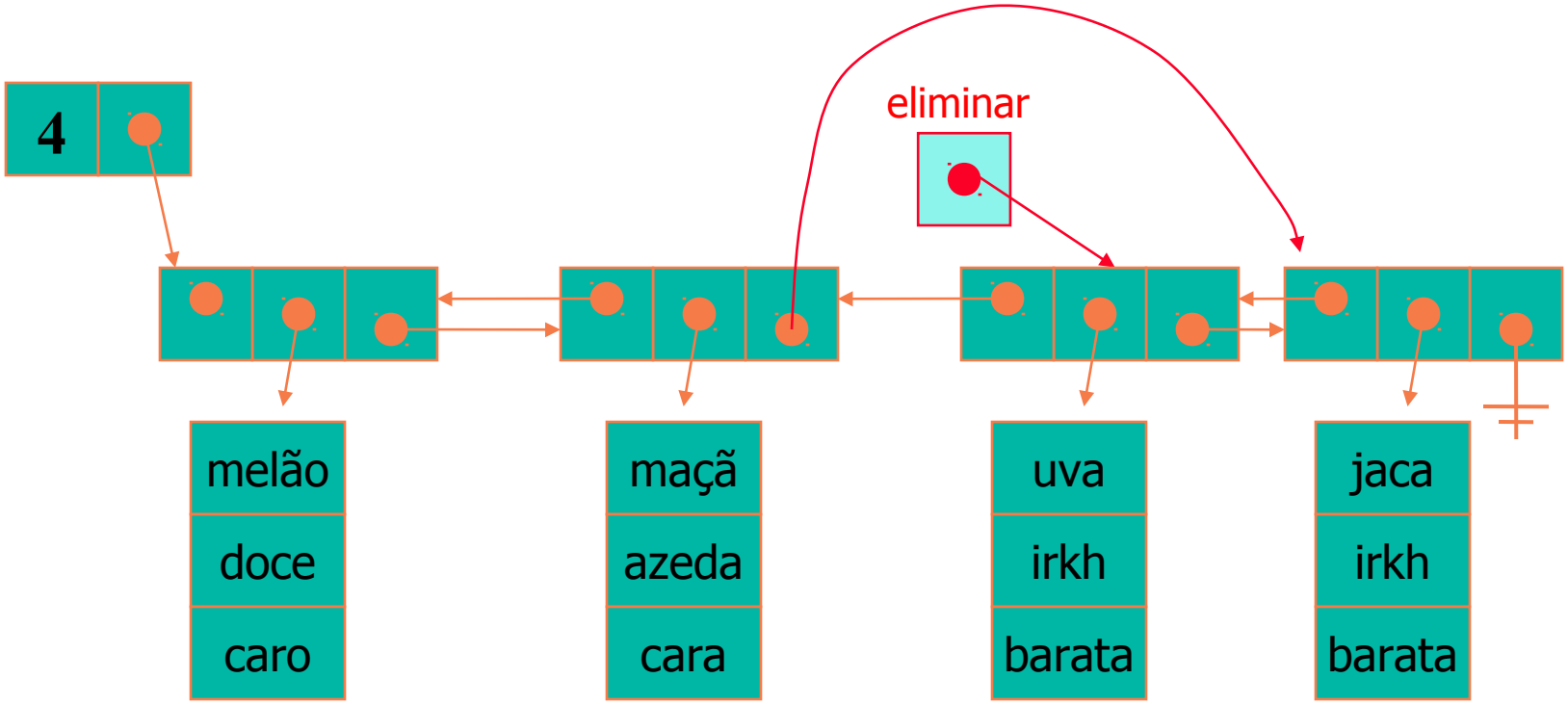
Algoritmo AdicionaPosiçãoListaDupla

Posições > 1



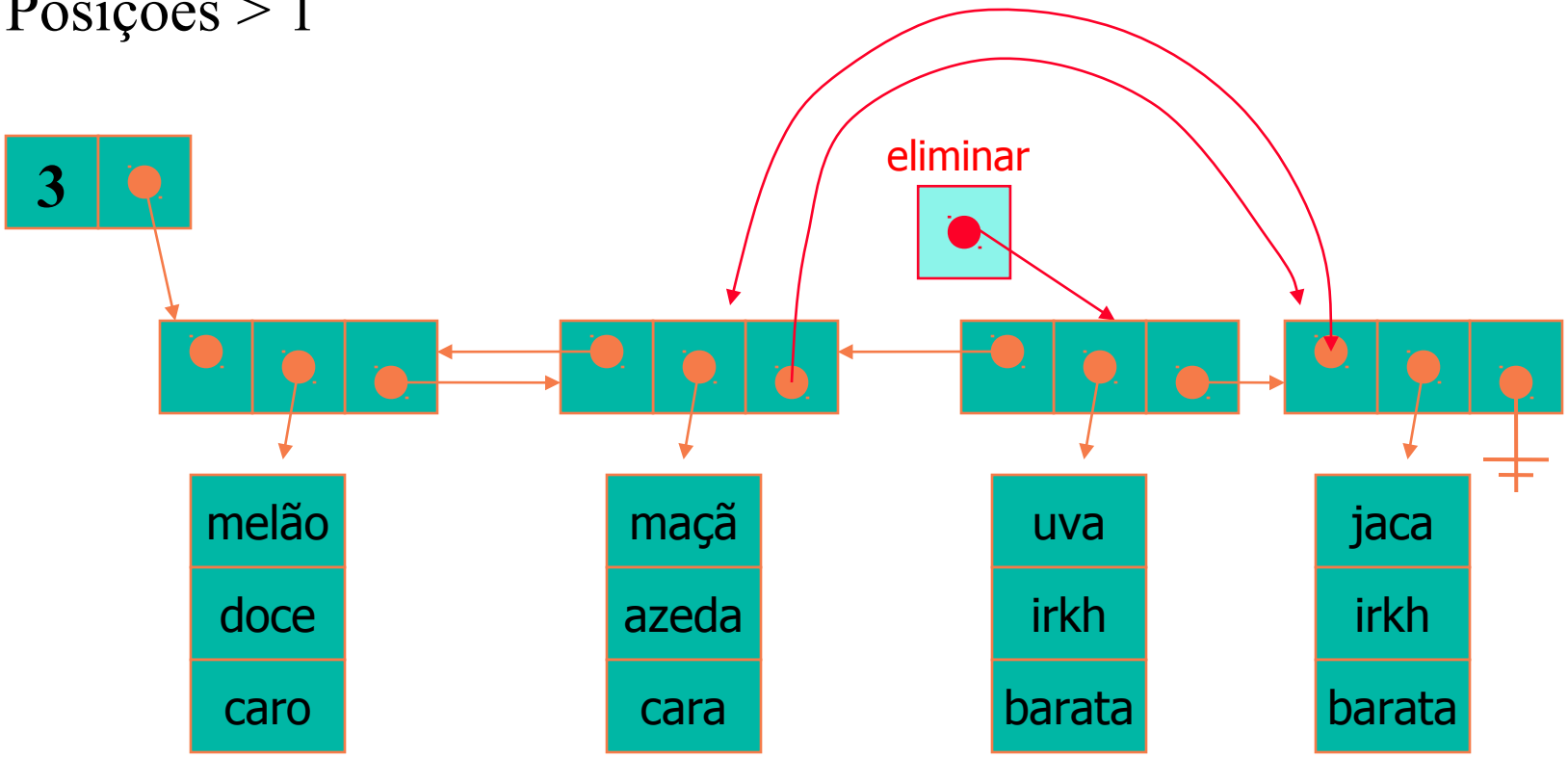
Algoritmo RetiradaPosiçãoListaDupla

Posições > 1



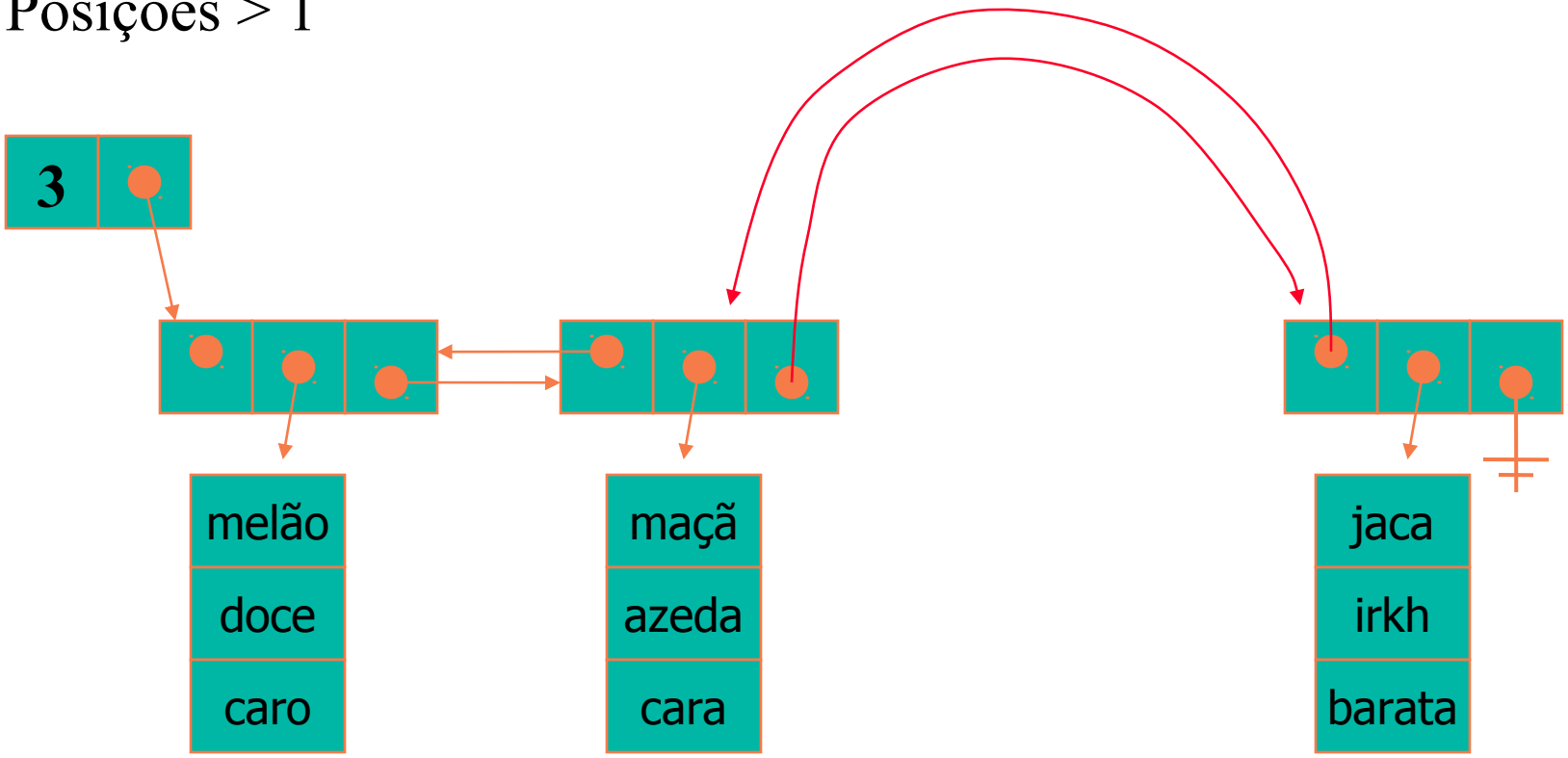
Algoritmo RetiradaPosiçãoListaDupla

Posições > 1



Algoritmo RetiradaPosiçãoListaDupla

Posições > 1



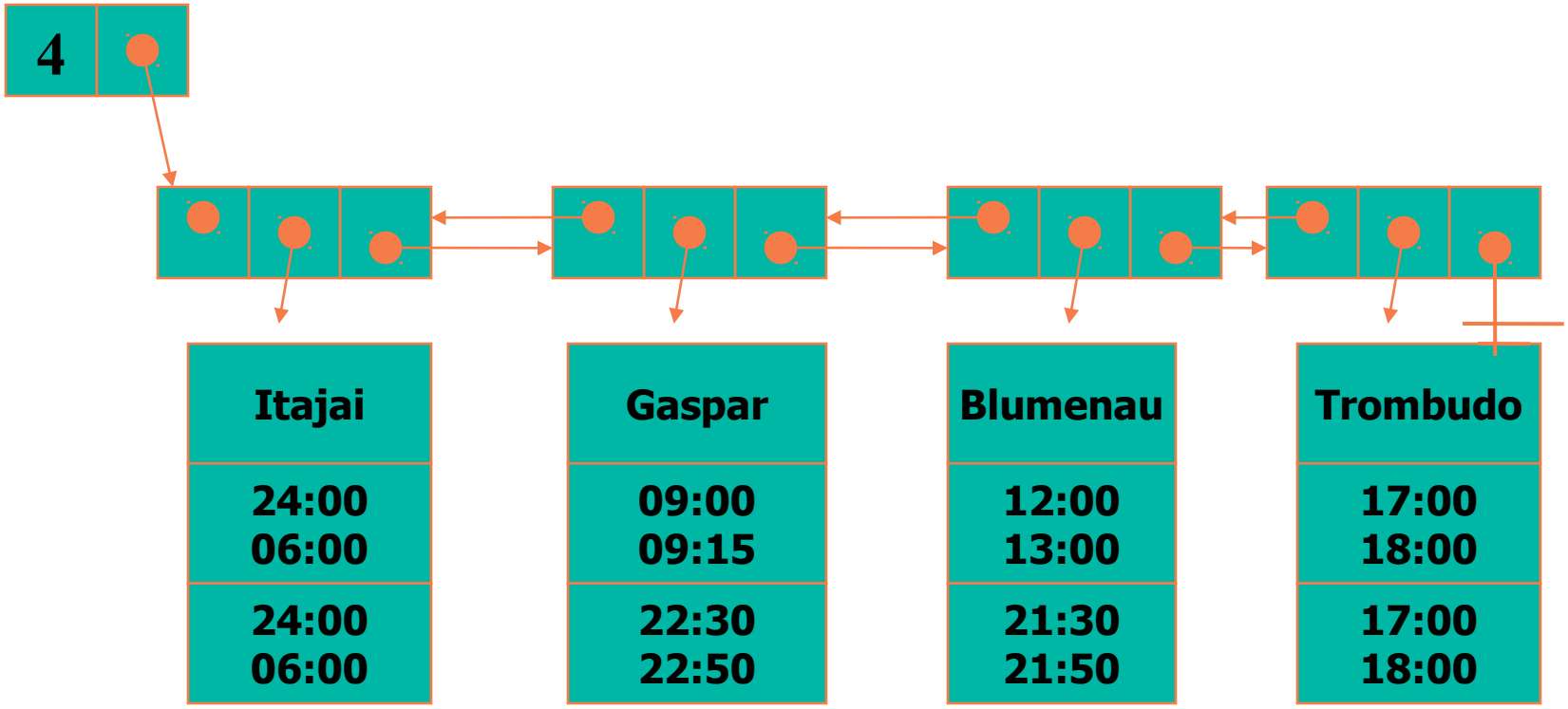
Algoritmo AdicionarEmOrdemDupla

- Idêntico à lista encadeada.
- Procedimento:
  - Necessitamos de uma função para comparar os dados (maior)
  - Procuramos pela posição onde inserir comparando dados.
  - Chamamos **`adicionaNaPosiçãoDuplo`**.
- Parâmetros:
  - O dado a ser inserido.
  - ListaDupla.

Exercício

- Implemente o TAD Lista Duplamente Encadeada com funções para todas as suas operações.
- Implemente um Módulo aplicativo programa principal que gerencie uma lista representando um itinerário de um trem. O programa deverá:
  - Utilizar um tipo info que represente uma estação de trem tendo:
    - Nome da estação
    - Hora de chegada e de saída
    - Hora de chegada e saída na volta.
  - Aceitar o cadastro de uma estação. O programa coloca a estação na posição por ordem de hora de saída na ida.
  - O programa deverá informar ao usuário dados sobre uma parada solicitada pelo usuário por nome:
    - Quando o trem chega na ida e na volta, qual a estação anterior e qual a estação que vem depois.

Exercício


4
Itajai
24:00
06:00
24:00
06:00
Gaspar
09:00
09:15
22:30
22:50
Blumenau
12:00
13:00
21:30
21:50
Trombudo
17:00
18:00
17:00
18:00