

Estrutura de Dados

Prof. Rafael Nunes

Listas

Parte 1

O que é uma Lista?

Lista

Estruturas de armazenamento
de dados linear



Prof. então vetores podem ser considerados Listas?

Prof. então vetores podem ser considerados Listas?

Sim! – Podemos!

... Porém temos algumas desvantagens

Vetores como Listas - Desvantagens

vetor[100]; // 100 posições estáticas

- **Tamanho** do vetor é **fixo**.
- Não existe a possibilidade de se aumentar durante a **execução**.
- As operações de **inserção/remoção** dispendiosas (em termos de processamento)

Tipos de Listas

Quais são?

Tipos de Listas

- Listas ***lineares***
 - estáticas
 - dinâmicas
- Como manipulá-las:
 - ***filas***
 - ***pilhas***

Tipos de operações

As operações básicas

Tipos de operações

- Operações básicas:
- ***Busca***
- ***Inclusão***
- ***Remoção***

Exercício

*Implementando uma lista
utilizando vetores*

Exercício

- Implemente um programa que simule o funcionamento de uma lista linear estática utilizando os protótipos das funções abaixo:

//Insere um caractere no vetor vet

void Insere(char x, char *vet, int pos);

//Busca um caractere no vetor vet

int Busca(char x, char *vet);

//Remove a primeira posição do vetor vet

void Remove(...)

Listas Lineares

Estáticas

Listas Lineares

- Fila:
 - Definição:
 - “Uma fila é uma **lista linear** em que todas as inserções são realizadas em **um extremo da lista**, e todas as retiradas são realizadas **no outro extremo da lista**” (ZIVIANI, 2004, p. 81)

Listas Lineares

- Fila:
 - estrutura de dados linear em que a *inserção* e a *remoção* de elementos de uma seqüência se faz por *extremidades opostas*, geralmente designadas por *cabeça (inicio)* e *cauda (fim)* da fila
- uma fila pode ser considerada como uma *restrição de lista*

Listas Lineares

- Fila:
- Comportamento
 - FIFO – *First In First Out* = Fila
 - “Primeiro a entrar, primeiro a sair” (CORMEM et al., 2002, p. 163)

Listas Lineares Estáticas - LLE

Comportamento

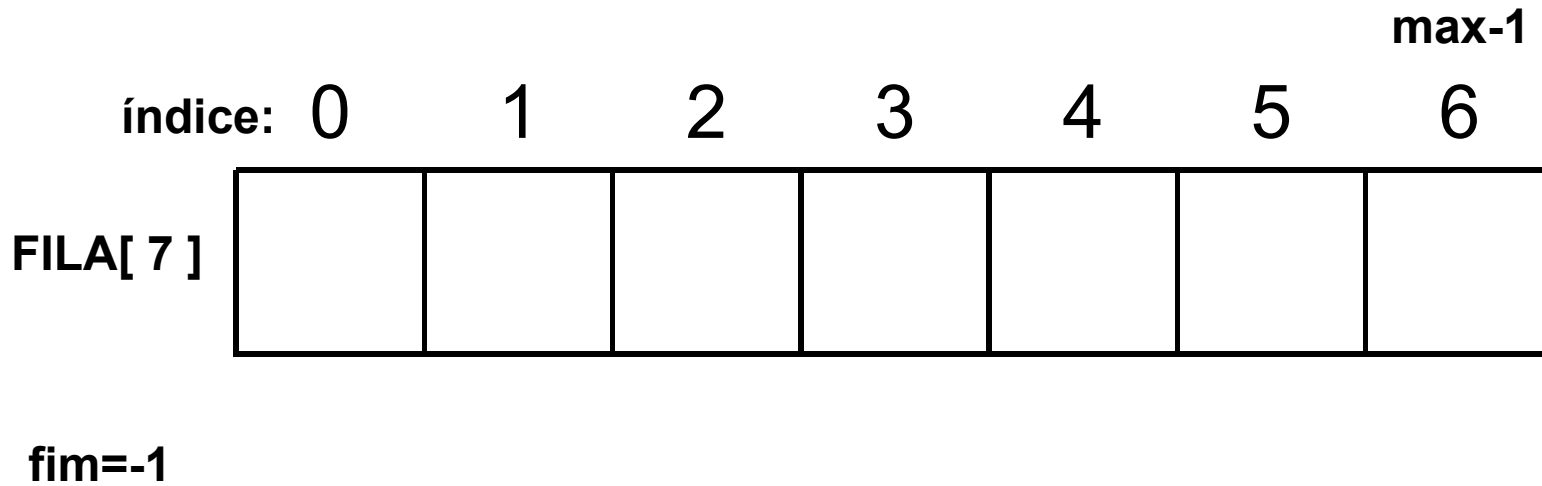
LLE - Comportamento

- Inserir a cadeia de caracteres “RAFAEL” na Lista abaixo mantendo as características de uma string
- Manipular a *lista* utilizando *fila*



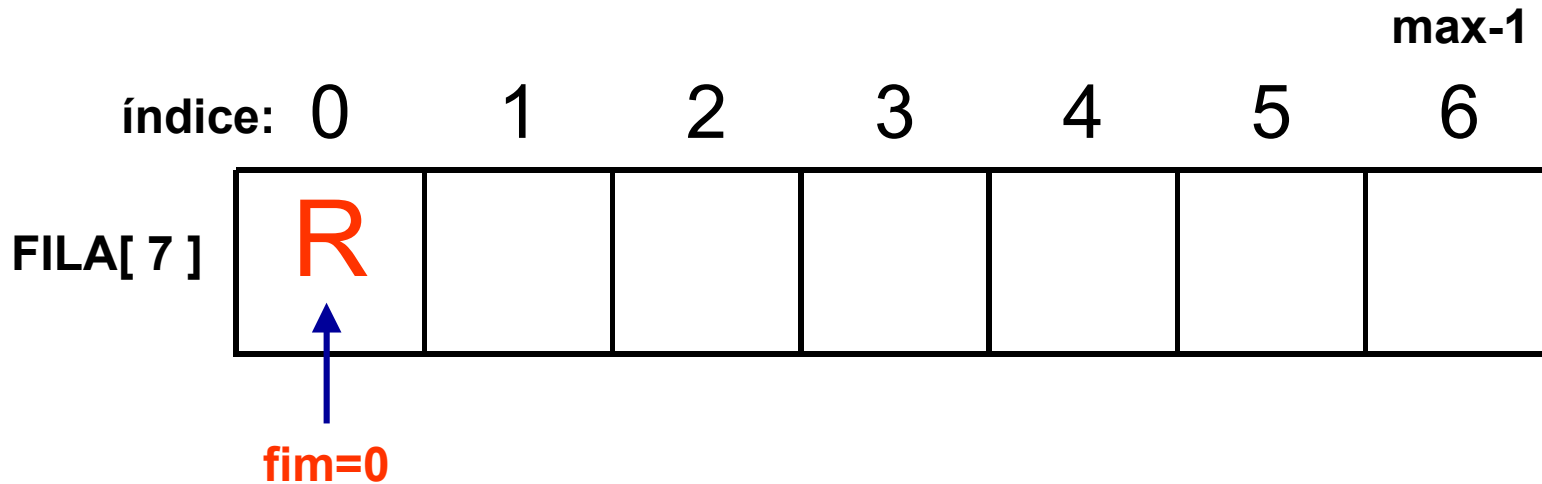
LLE - Comportamento

INSERE (R)



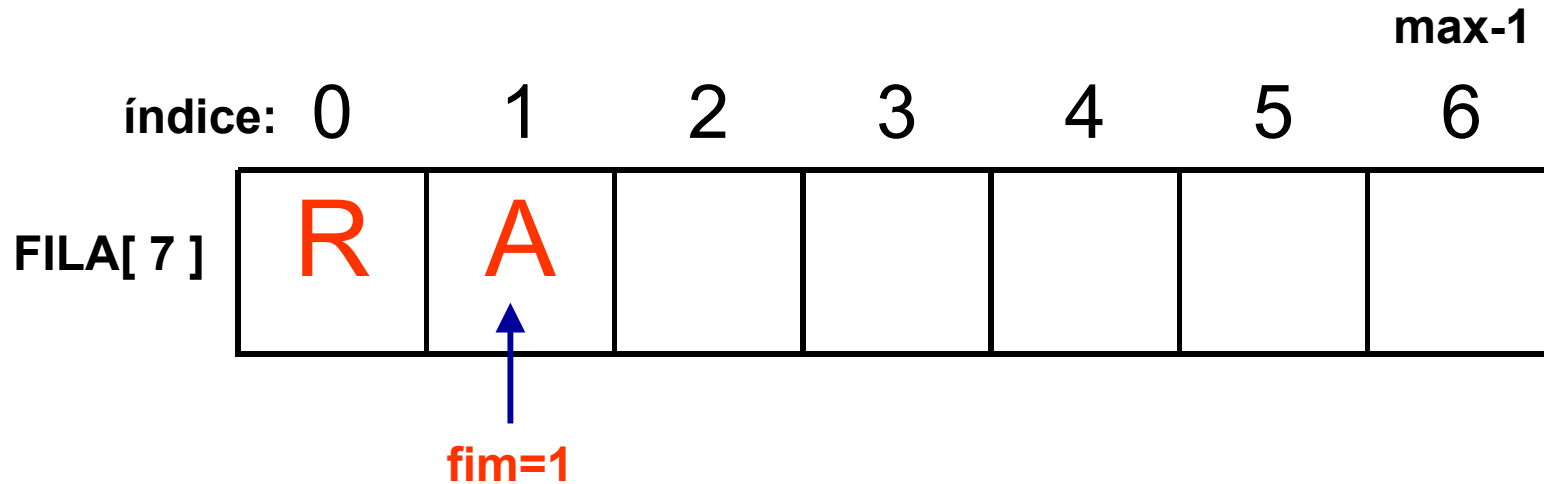
LLE - Comportamento

INSERE (R)



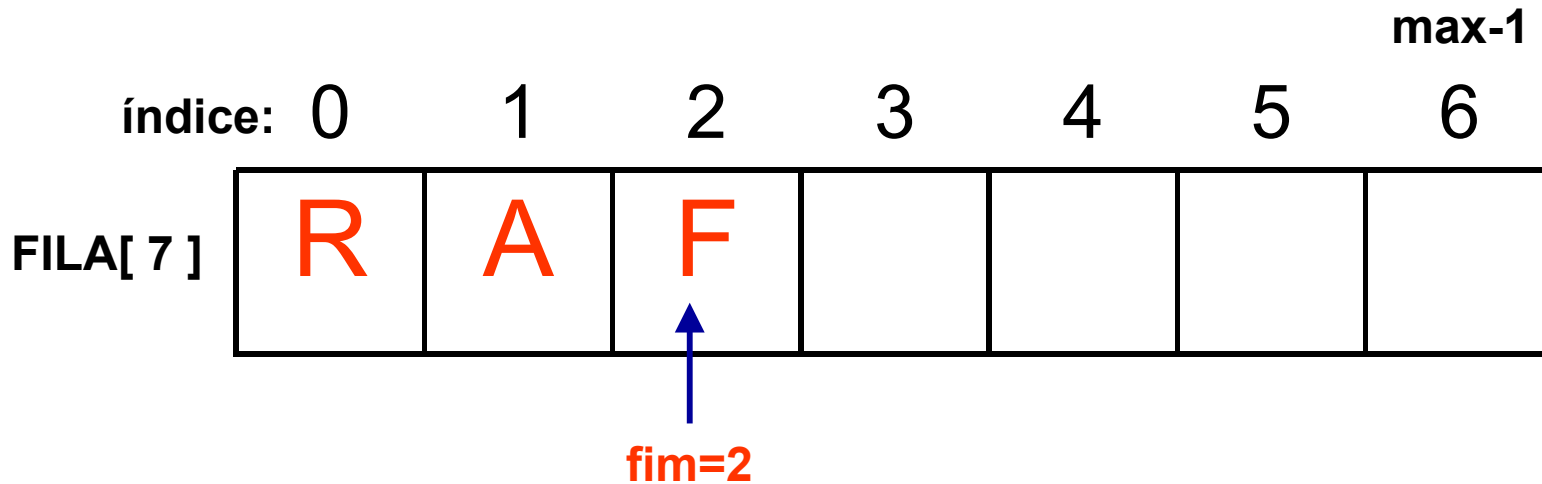
LLE - Comportamento

INSERE (A)



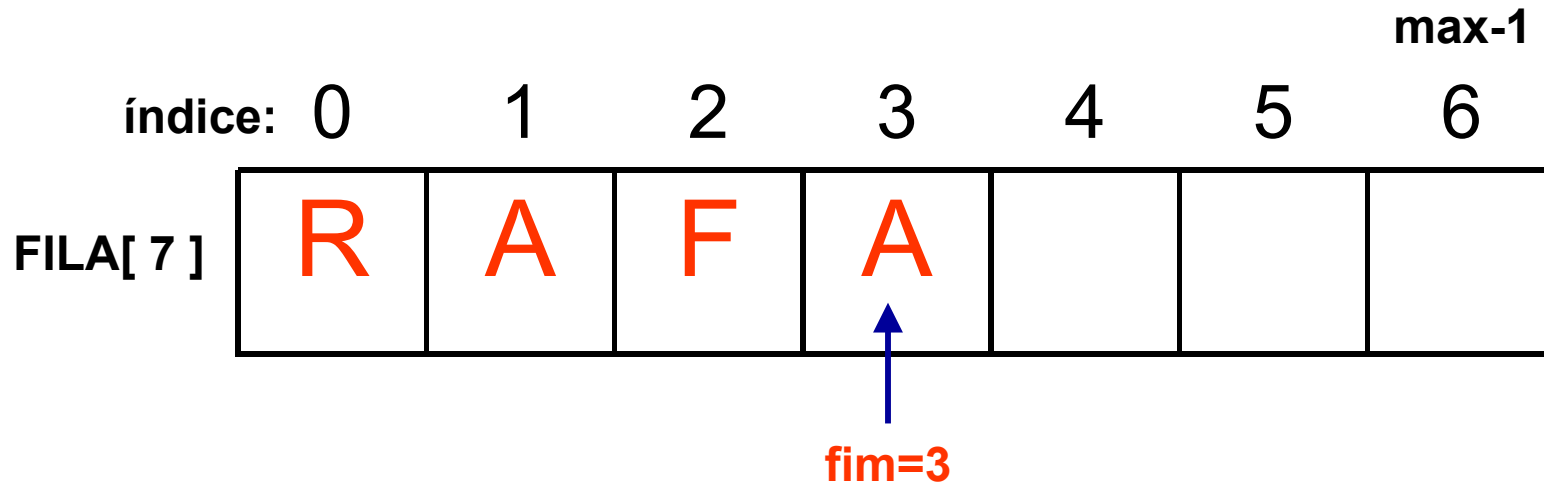
LLE - Comportamento

INSERE (F)



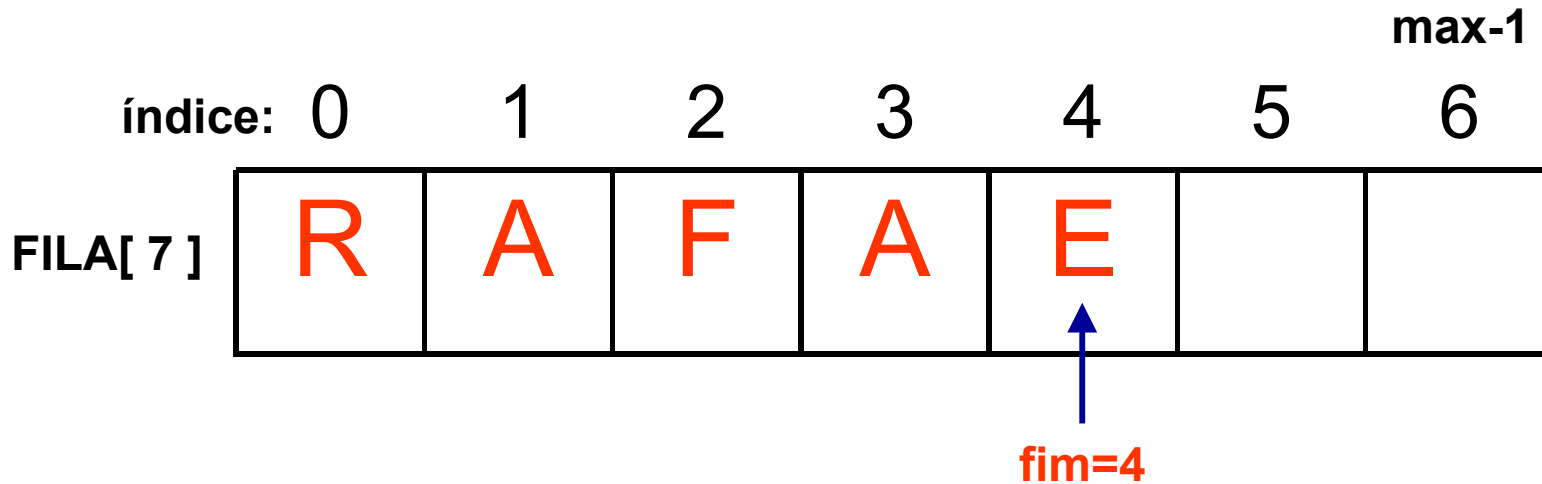
LLE - Comportamento

INSERE (A)



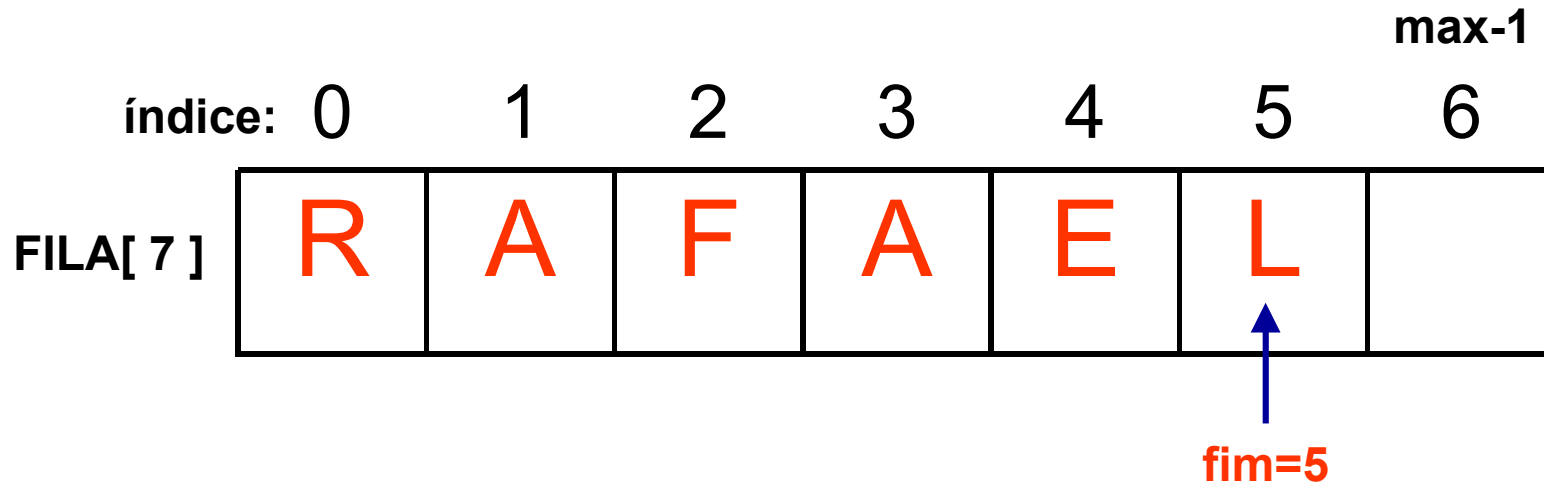
LLE - Comportamento

INSERE (E)



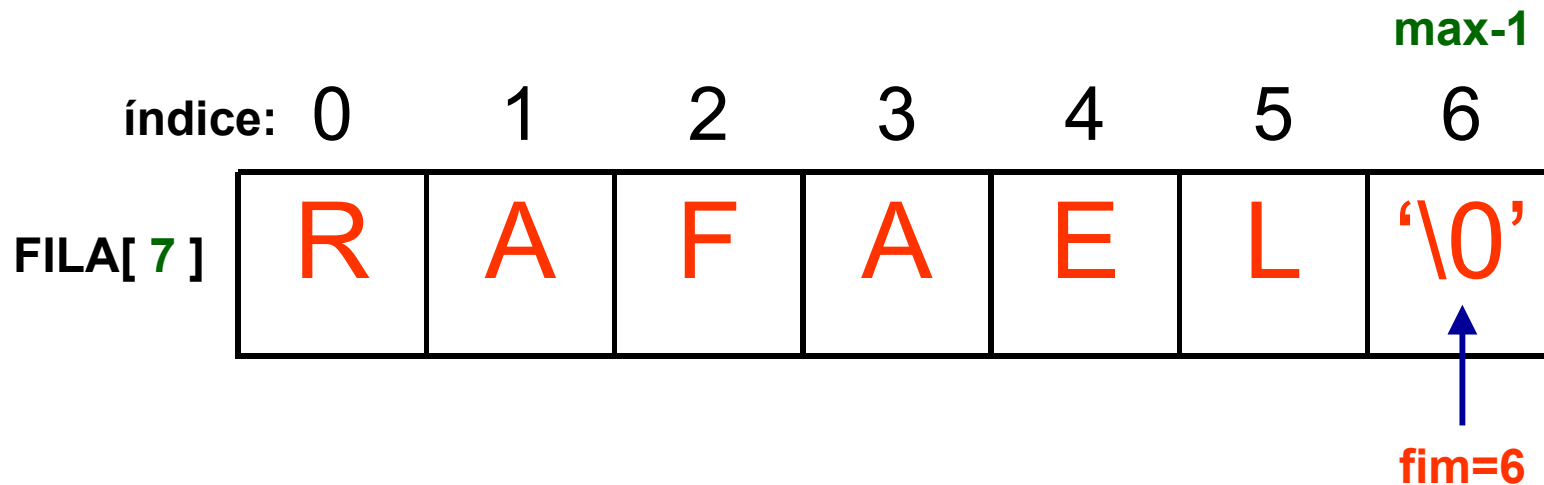
LLE - Comportamento

INSERE (L)



LLE - Comportamento

NÃO ESQUECER DO '\0'

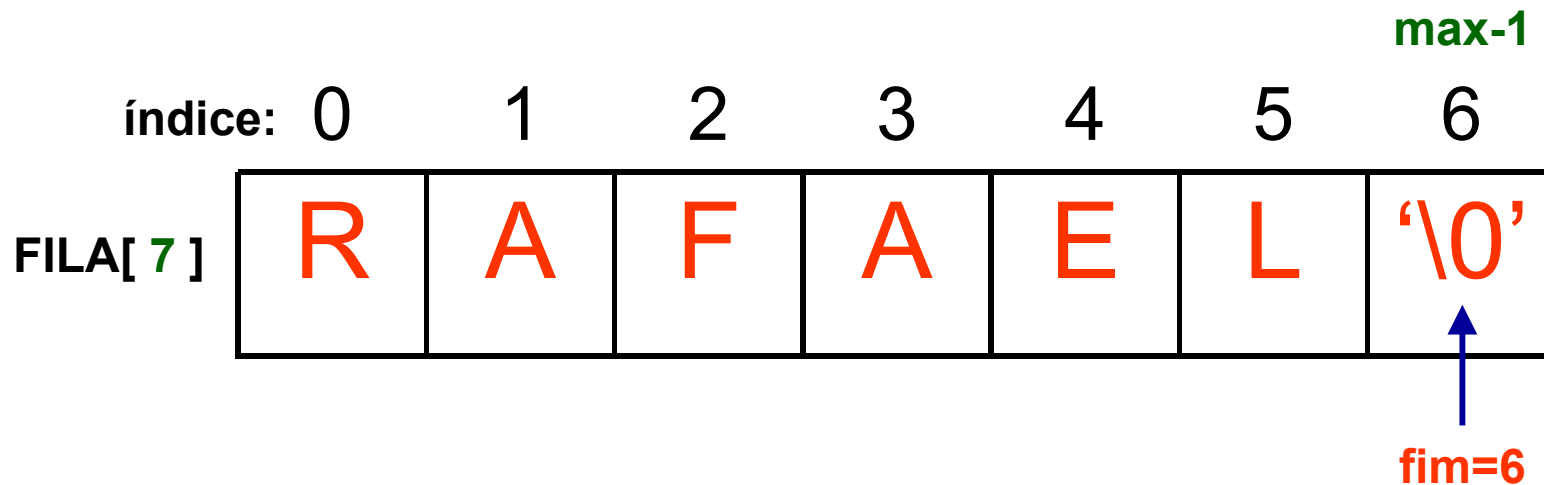


Remoção

*Retirar elemento da fila
sempre do Início*

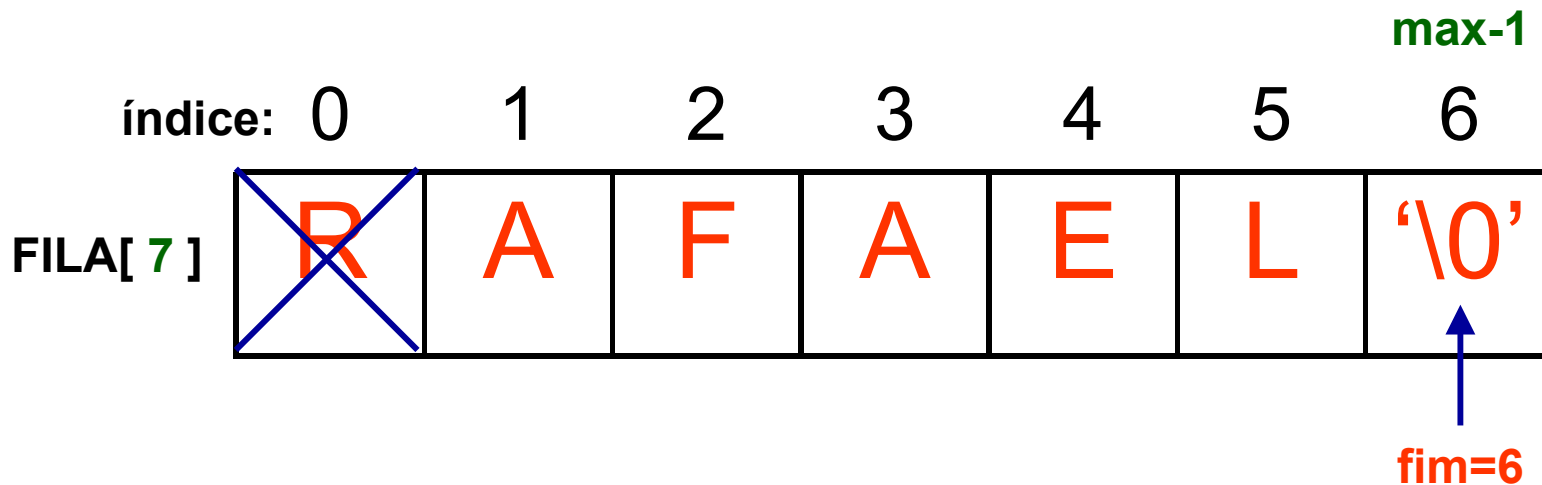
LLE - Comportamento

REMOVE – O primeiro que entrou



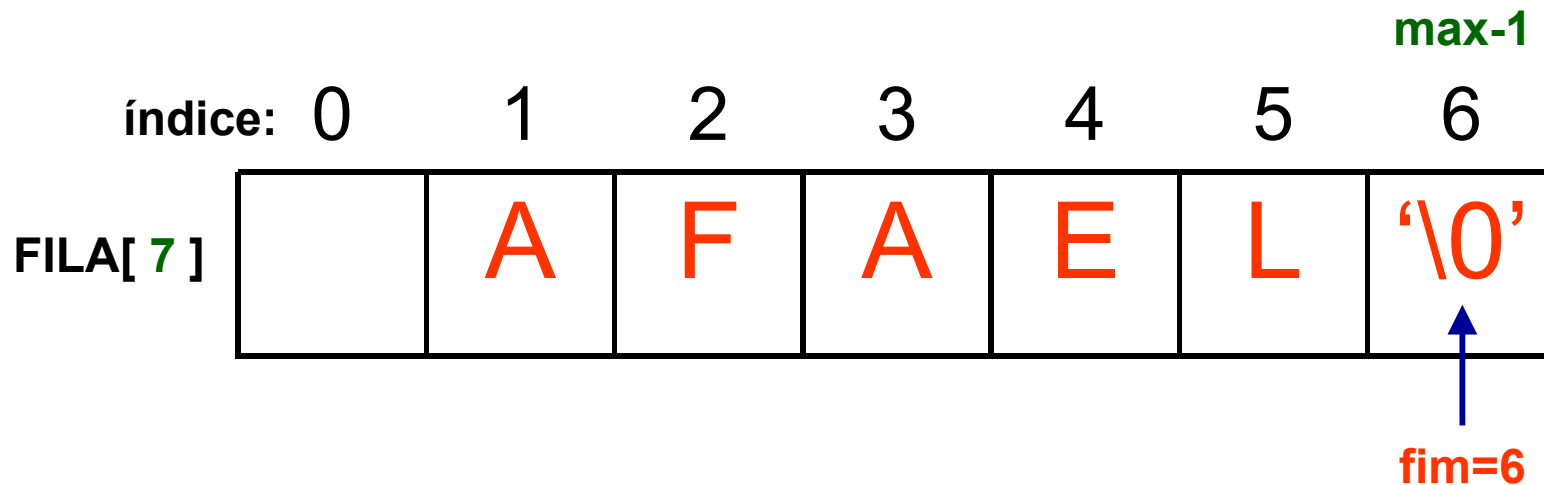
LLE - Comportamento

REMOVE – O primeiro que entrou



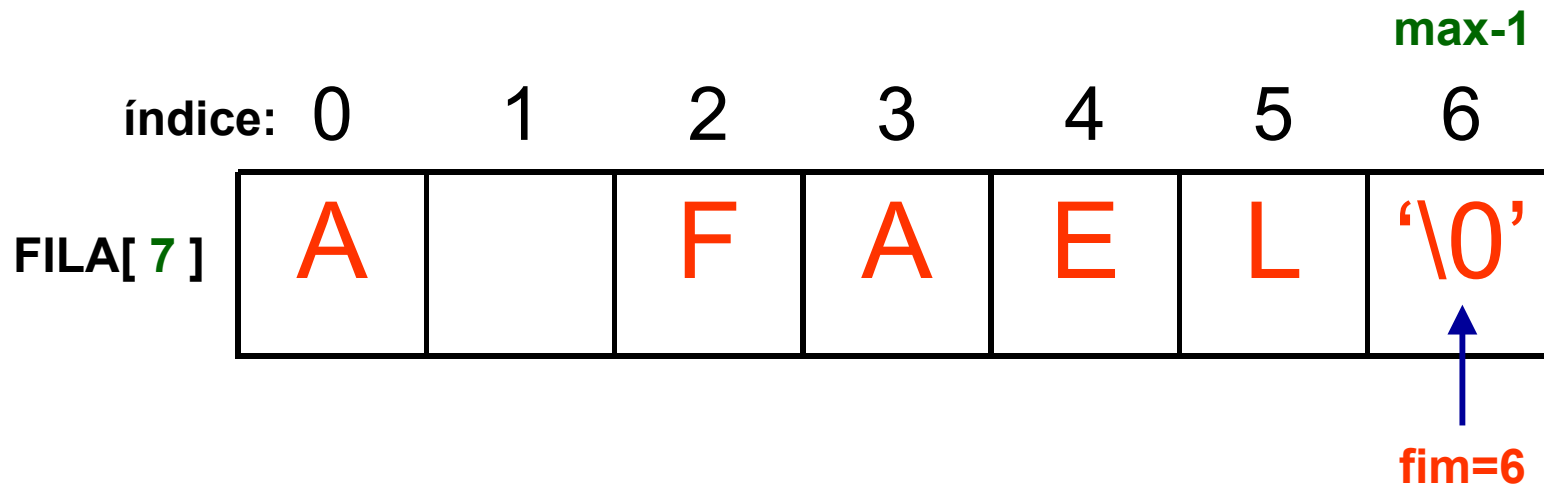
LLE - Comportamento

REMOVE – O primeiro que entrou



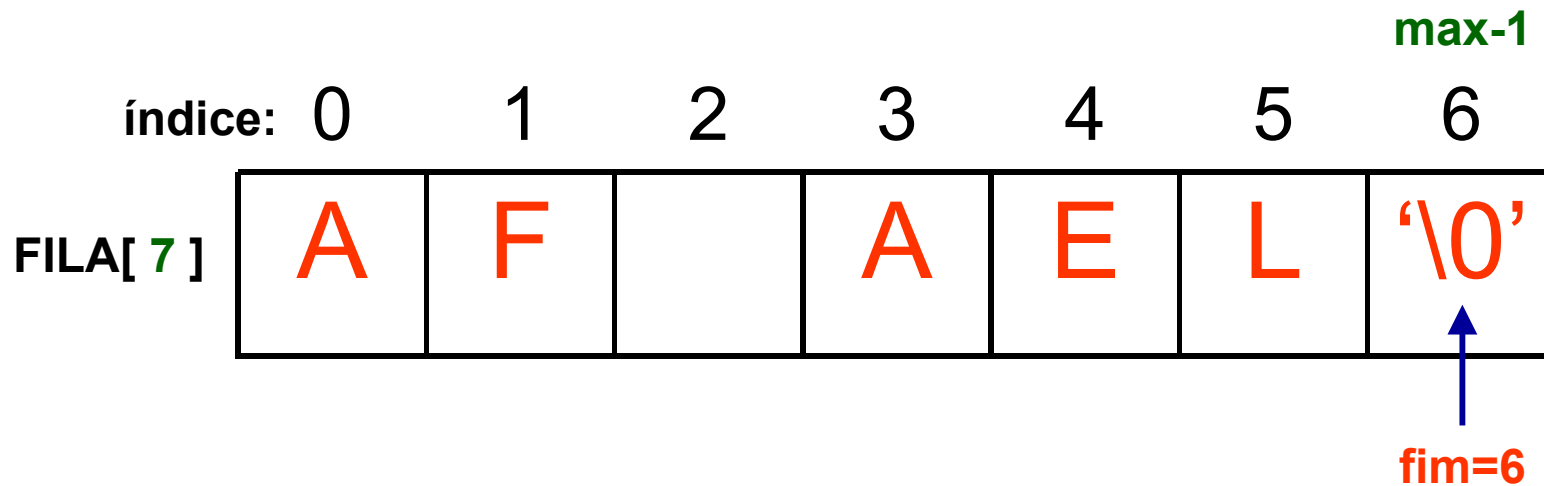
LLE - Comportamento

REESTRUTURA A LISTA



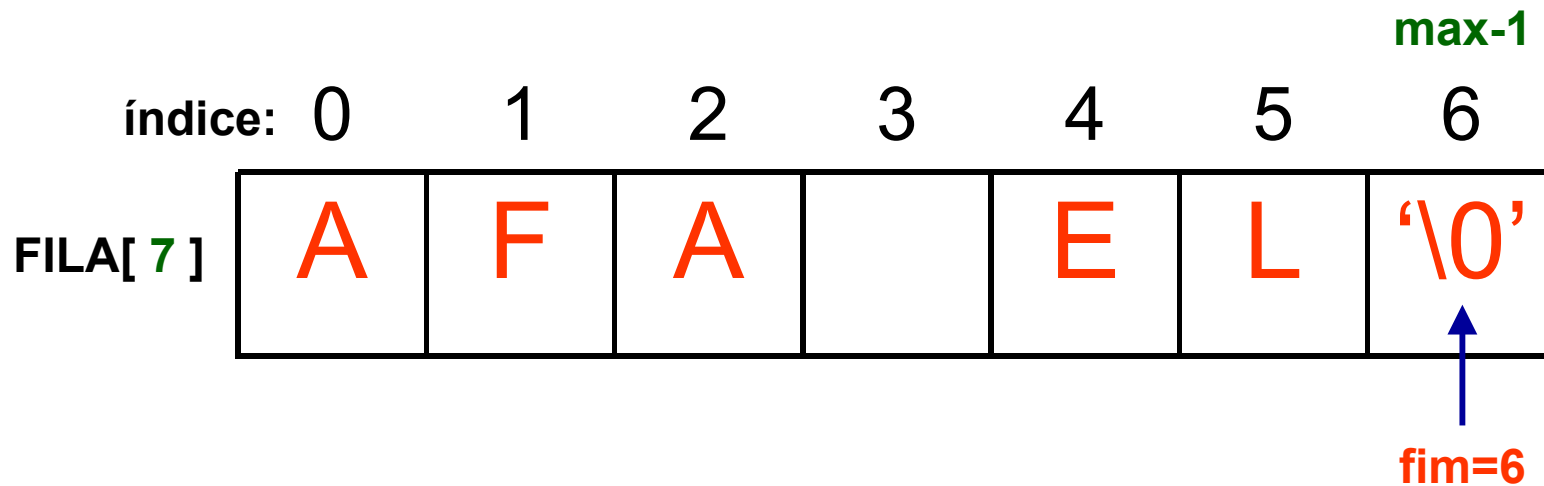
LLE - Comportamento

REESTRUTURA A LISTA



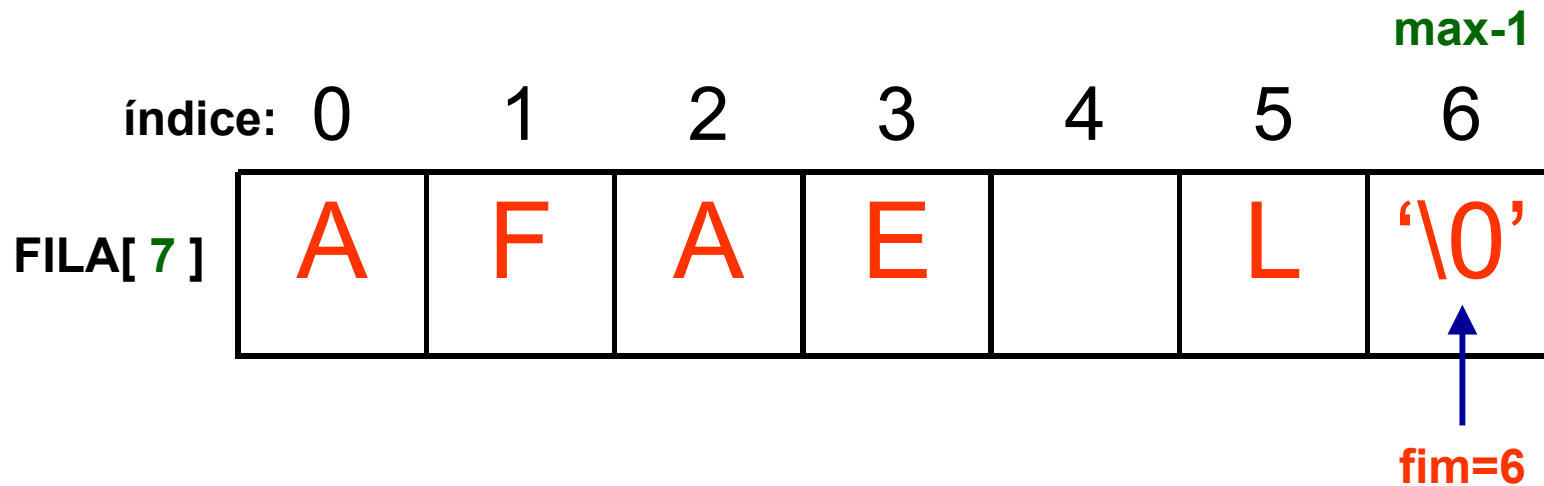
LLE - Comportamento

REESTRUTURA A LISTA



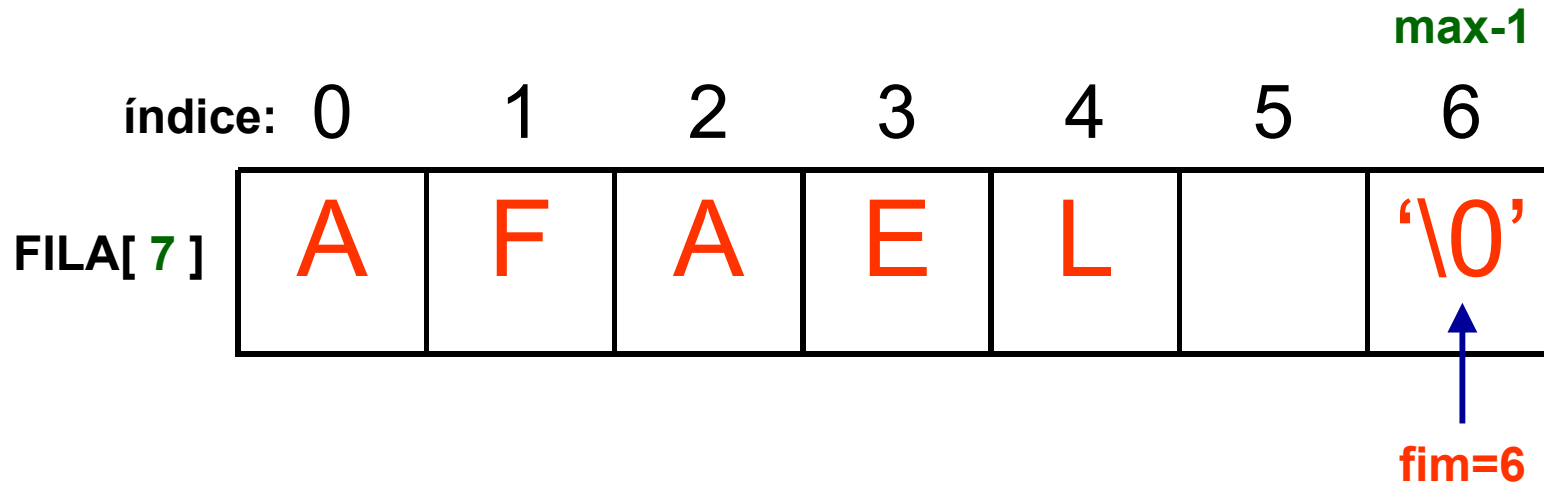
LLE - Comportamento

REESTRUTURA A LISTA



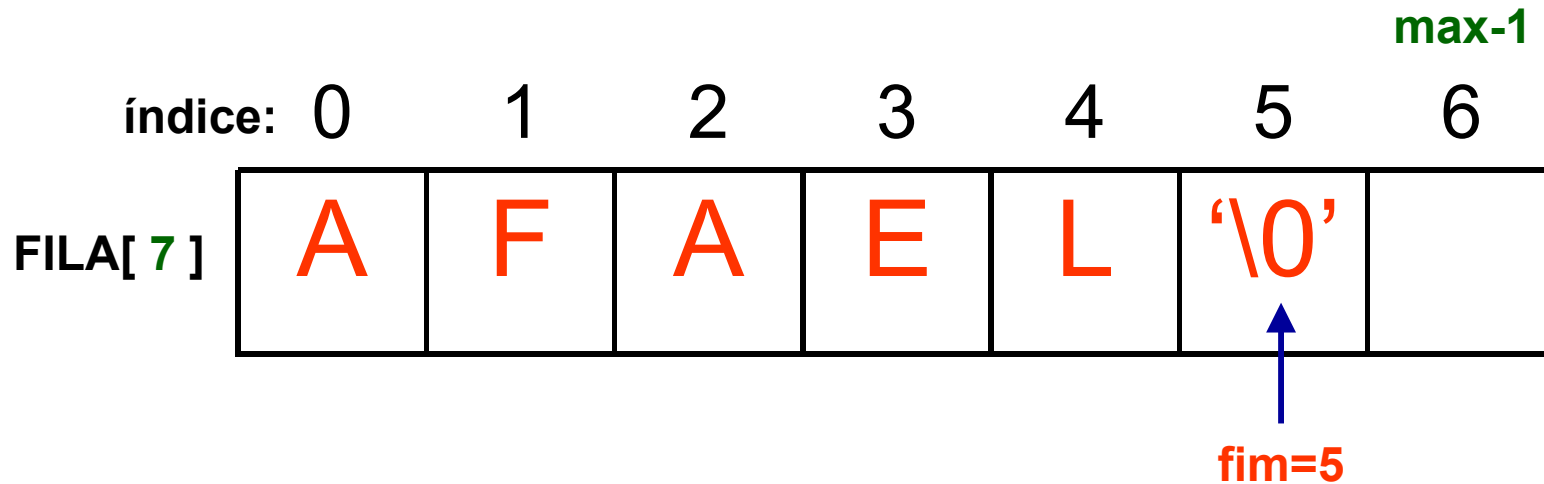
LLE - Comportamento

REESTRUTURA A LISTA



LLE - Comportamento

REESTRUTURA A LISTA



LLE - Comportamento

- Ao **INSERIR** (ENFILEIRAR) verificar se **fila está cheia** ($\text{fila} == \text{max} - 1$):
 - se fila cheia, imprimir mensagem “fila cheia”;
 - senão, INSERE elemento e atualiza fim da fila.
- Ao **REMOVER** (DESENFILEIRAR) verificar se **fila está vazia** ($\text{fim} == -1$):
 - se fila vazia, não REMOVA. Imprima mensagem “fila vazia”;
 - senão, REMOVA, mostre o elemento removido e movimente os elementos das posições $i+1$ (posição posterior) para i (posição anterior)
 - $i = (0 \dots \text{max} - 1)$

Nas listas precisamos *garantir*
a ordem dos elementos

Como garantir a ordem dos elementos?

- Pode ser feita de **duas formas**:
 - Garantir a precedência dos elementos pela **contigüidade física** na memória utilizando os índices de um arranjo (vetores) – **Já visto!**
 - Garantir a precedência dos elementos utilizando **apontadores**, encadeamento

Listas Lineares

Dinâmicas

Listas Lineares Dinâmicas

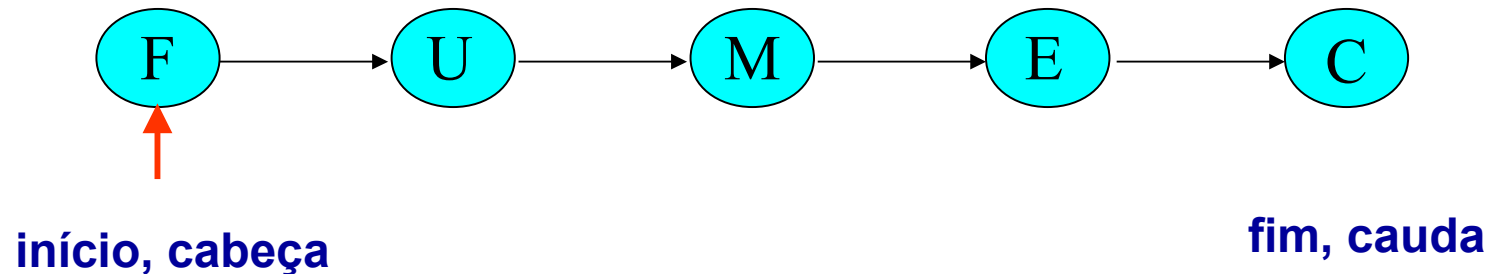
- Podem crescer e diminuir ***dinamicamente***
- O ***tamanho máximo*** não precisa ser conhecido a priori, ou seja, não precisamos ***definir o tamanho*** estaticamente
- Provêem flexibilidade ***permitindo*** que os itens sejam ***rearranjados*** eficientemente
 - *Desvantagem*: perda no tempo de acesso a qualquer item arbitrário da lista

Tipos de Listas

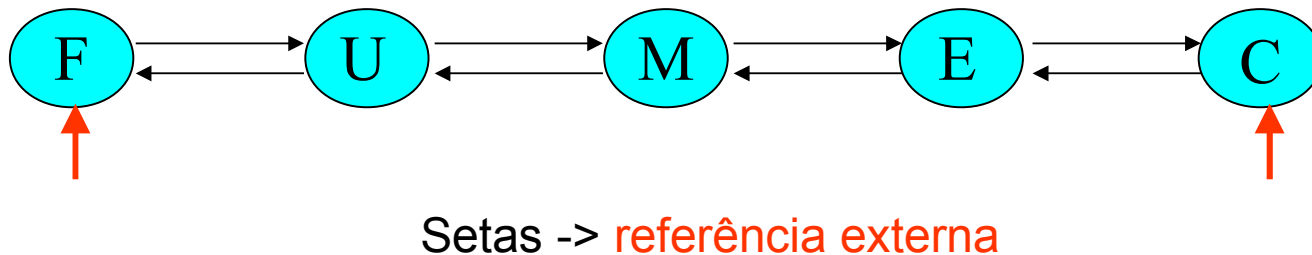
Lineares Dinâmicas

Tipos de Listas Lineares Dinâmicas

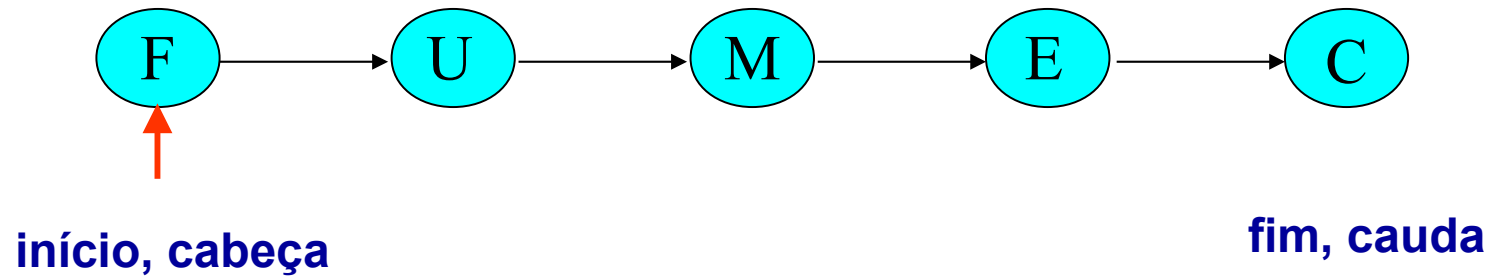
- Encadeadas -> Simplesmente Ligadas



- Duplamente encadeadas -> Duplamente Ligadas



Listas Encadeadas



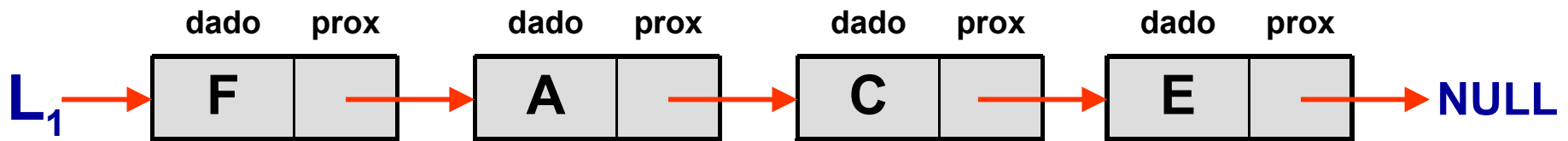
Listas Encadeadas

- É um conjunto de itens organizados, como um **array** (lista seqüencial)
 - em um array a **organização é implícita** (pela posição)
 - em uma lista encadeada a seqüência de elementos é especificada **explicitamente**, onde cada elemento **contém um ponteiro** para **o próximo da lista**.

Exemplo

Lista simplesmente encadeada

Lista Simplesmente Encadeada



Lista Simplesmente Encadeada

- No exemplo:
 - um **elemento** é identificado por uma letra
 - cada elemento é representado pela estrutura, **info + prox**
 - Cada link é representado por uma seta
- Detalhes que devem ser considerados:
 - todo elemento possui um **ponteiro**
 - o ponteiro do último elemento tem que especificar algum **tipo de próximo** (aponta para si próprio ou NULL)

Exercício

*Implementar o caso mais simples
de Lista Encadeada*

Exercício

- Implemente o caso mais simples de lista encadeada
- A inserção dos elementos deverá ser feita no início da lista
- As seguintes funções deverão ser implementadas
- `inicializa()`, `insere()`, `imprime()`

Até a próxima...