

LISTAS RESTRITAS (FILA)

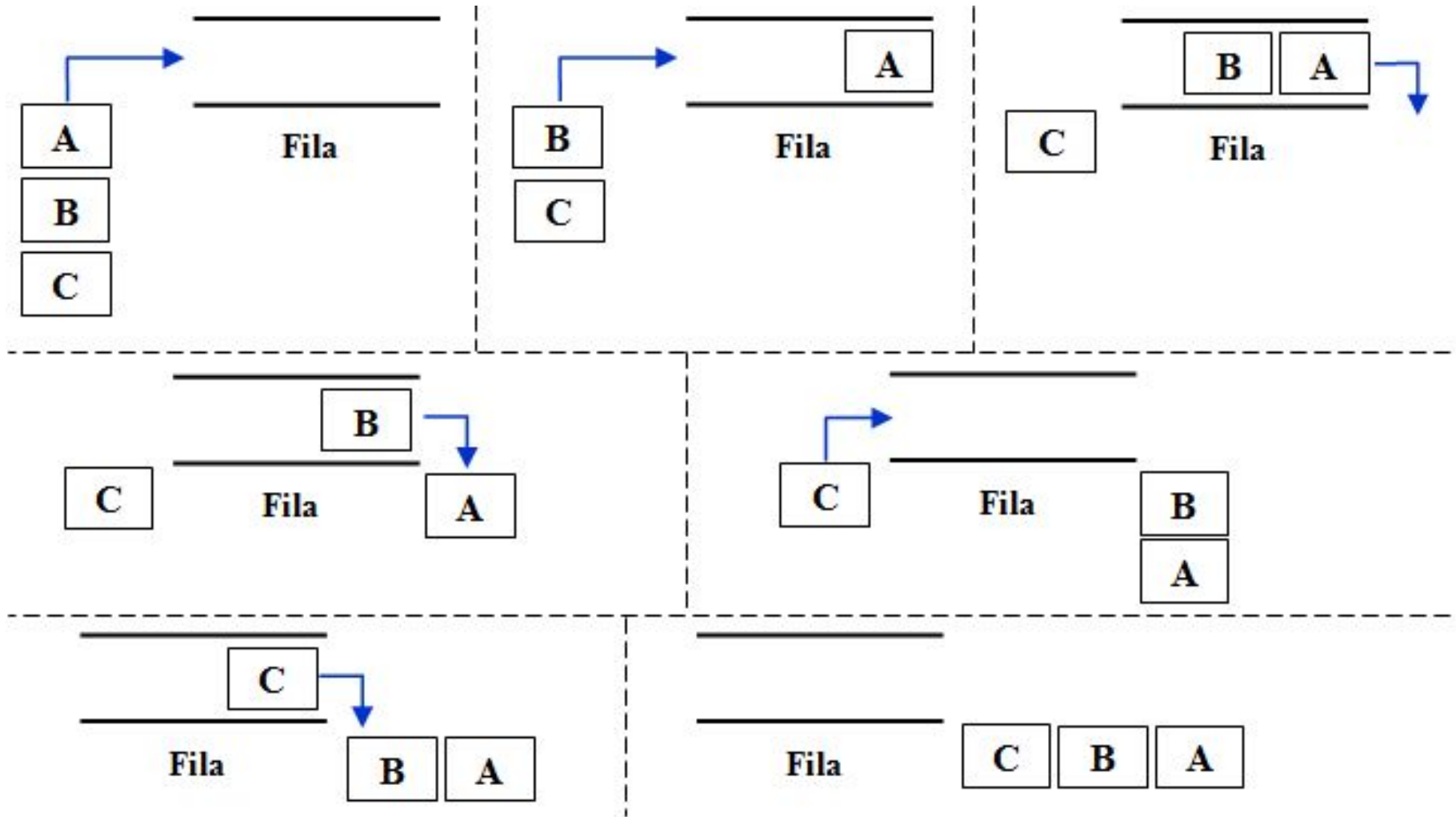
Prof. Alberto Costa Neto

LISTAS RESTRITAS: DEFINIÇÃO

- **Definição:** É uma lista na qual:
 - **Inserções** são feitas em uma extremidade chamada "**cauda**" ou "**fundo**",
 - **Remoções** são feitas na outra extremidade, chamada "**cabeça**" ou "**frente**".
- Numa fila, **o primeiro a entrar é o primeiro a sair**.
- Esta política de acesso é denominada **FIFO** ("**First In, First Out**")

FILA

- Graficamente temos:



OPERAÇÕES

Inicializar	Cria uma fila vazia.
Inserir	Insere um elemento no fundo (cauda) da fila.
Retirar	Retira um elemento que está na frente (cabeça) da fila.
Frente	Retorna o elemento que está na frente (cabeça) da fila.
Vazia	Indica se a fila está vazia.

FORMAS DE IMPLEMENTAÇÃO DE UMA FILA

Há basicamente 2 formas de implementar uma Fila:

- **Usando uma lista encadeada:**

- O elemento **inserido** é colocado **no fim da lista (cauda)**.
- A **retirada** é feita no **início (cabeça)**.

- **Usando um array:**

- Há duas variáveis **(Início e Fim)** que **indicam as extremidades** da fila (cabeça e cauda).
- Ao **inserir**, o novo elemento é **colocado na posição Fim + 1**.
- Ao **retirar** um elemento, isto é feito **na posição Início**.

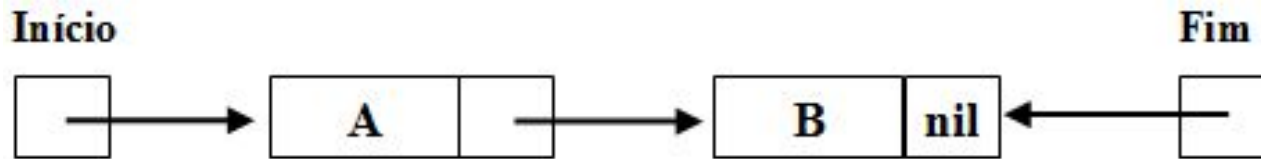
FILA BASEADA EM
LISTA ENCADEADA

FILA BASEADA EM LISTA ENCADEADA

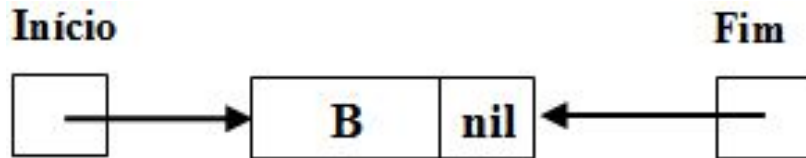
- **Inicializar** faz com que o início e o fim da fila recebam NULL (nil) e o tamanho seja zero.
- **Inserir** coloca a informação em um novo nó e faz com que nó no fim aponte para ele. Também ajusta o apontador fim para o novo nó.
- **Retirar** obtém a informação contida no nó do início da fila e retirar este nó da lista.
- **Frente** obtém a informação contida no nó que está no início da fila, sem retirá-lo de lá.

FILA BASEADA EM LISTA ENCADEADA

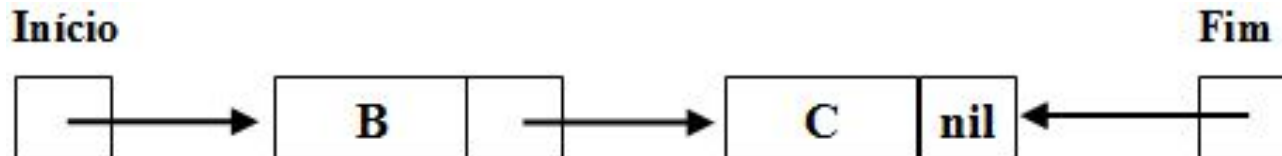
Inserindo A e B:



Retirando o elemento da frente (A):



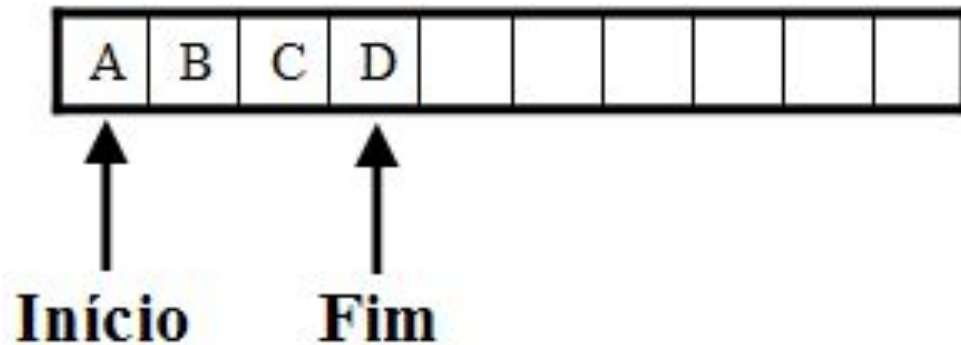
Inserindo C:



FILA BASEADA EM ARRAY

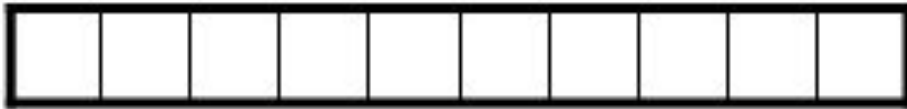
FILA BASEADA EM ARRAY

- A implementação de uma fila é semelhante à de uma lista seqüencial onde:
 - **Inserções** são feitas no final da lista;
 - **Remoções** no início da lista.



FILA BASEADA EM ARRAY: FUNCIONAMENTO

Fila vazia:



Inserindo A e B:



Retirando o elemento da frente (A):


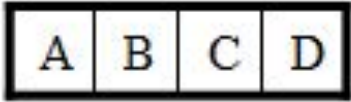
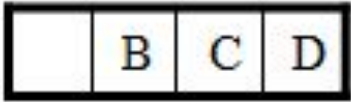
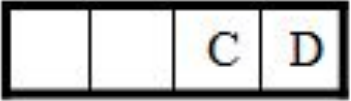
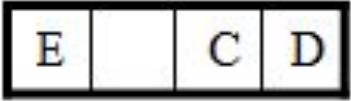
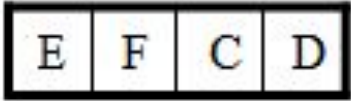


Inserindo C:



FILA BASEADA EM ARRAY (OTIMIZAÇÃO)

- Para evitar que a cada retirada de um item da fila seja necessário deslocar todos os elementos do array para a "esquerda", usamos três variáveis: **Início**, **Fim** e **Tamanho**.

 <p>Início = 0 Fim = 0 Tamanho = 0</p>	 <p>Início = 1 Fim = 4 Tamanho = 4</p>	 <p>Início = 2 Fim = 4 Tamanho = 3</p>
 <p>Início = 3 Fim = 4 Tamanho = 2</p>	 <p>Início = 3 Fim = 1 Tamanho = 3</p>	 <p>Início = 3 Fim = 2 Tamanho = 4</p>

FILA BASEADA EM ARRAY (INSERÇÃO) EM PASCAL

```
if tamanho < MAX then
```

```
    begin
```

```
        Fim = (Fim mod MAX) + 1;
```

```
        itens[Fim] = item;
```

```
        tamanho = tamanho + 1;
```

```
    end
```

```
else
```

```
    begin
```

```
        // Fila cheia
```

```
    end
```

FILA BASEADA EM ARRAY (RETIRADA) EM PASCAL

```
if tamanho > 0 then
```

```
    begin
```

```
        item = itens[inicio];
```

```
        inicio = (inicio % Max) + 1;
```

```
        tamanho = tamanho - 1;
```

```
    end
```

```
else
```

```
    begin
```

```
        // Fila Vazia
```

```
    end
```

FILA BASEADA EM ARRAY (INSERÇÃO) EM C

```
if (tamanho < MAX)
{
    Fim = (Fim + 1) % MAX;
    itens[Fim] = item;
    tamanho = tamanho + 1;
}
else
{
    // Fila cheia
}
```

FILA BASEADA EM ARRAY (RETIRADA) EM C

```
if (tamanho > 0)
{
    item = itens[inicio];
    inicio = (inicio + 1) % Max;
    tamanho = tamanho - 1;
}
else
{
    // Fila Vazia
}
```