

## Estruturas de Dados com disciplina de acesso do tipo FILA COM PRIORIDADES

Conceito: **"FIFO" (First IN, First OUT) - dentro da prioridade**

O acesso aos dados acontece em duas extremidades:

:: **Início:** Retirada de elementos;

:: **Fim:** Inserir os elementos e deslocá-los até a posição que mantenha a ordem de prioridades.

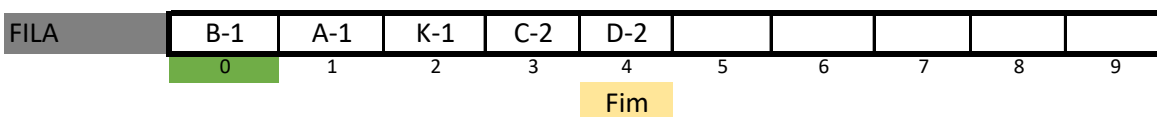
Exemplo com 3 prioridades:

:: Alta: 1

:: Média: 2

:: Baixa: 3

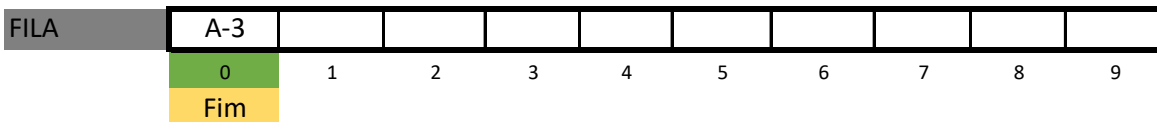
Exemplo de uso com INÍCIO fixado na posição "zero":



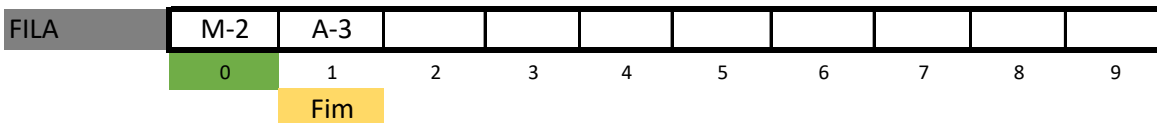
Exemplo de uso:

Inicializar -> Fim = -1

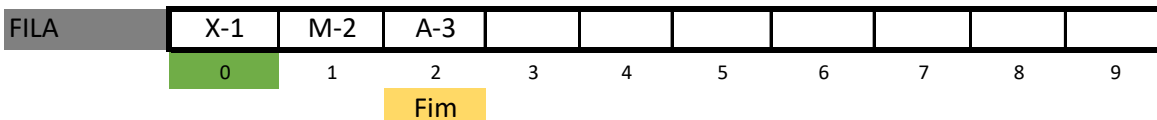
Inserir -> A-3



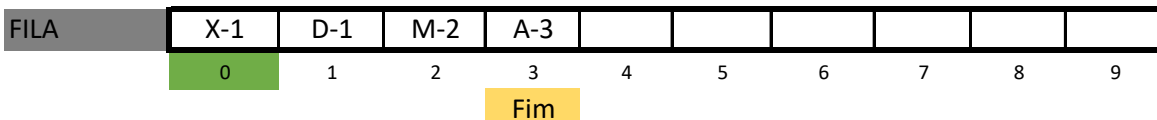
Inserir -> M-2



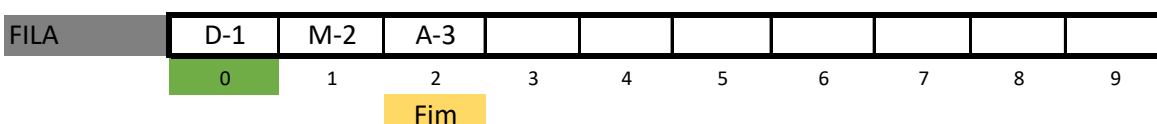
Inserir -> X-1



Inserir -> D-1



**Retirar ->** Obrigatoriamente, o elemento que está no INÍCIO da FILA, portanto (neste caso) o elemento (registro) da posição "zero" deve ser retornado.  
Com isso, a FILA deve ser deslocada para a esquerda e o FIM decrementado.



### ESTRUTURAS DE DADOS

```
#define MAXFILA 10
```

```
struct TpElemento
```

```
{  
    char Info;  
    int Prioridade;  
};
```

```
struct TpFila
```

```
{  
    int FIM;  
    TpElemento FILA[MAXFILA];  
};
```

### OPERAÇÕES ASSOCIADAS

```
void Inicializar(TpFila &F);
```

```
char FilaVazia(int fim);
```

```
char FilaCheia(int fim);
```

```
void Inserir(TpFila &F, TpElemento Elemento);
```

```
TpElemento Retirar(TpFila &F);
```

```
TpElemento ElementoInicio(TpFila F);
```

```
TpElemento ElementoFim(TpFila F);
```

```
void ExibirFila(TpFila F);
```



