

## TIPO ABSTRATO DE DADOS - FILA

Uma fila é uma lista com restrições de acesso: a inserção de um novo elemento somente pode ser feita em uma das extremidades da lista (denominada fim) e a remoção somente pode ser feita na outra extremidade (denominada inicio). Essas restrições configuram a política FIFO (first in, first out), o primeiro que entra é o primeiro que sai.

### Operações

Construtor: criarFilaVazia - operação que permite criar uma fila vazia.

Acesso:

acessarPrimeiro - devolve o primeiro elemento da fila

verificarFilaVazia - devolve true se a fila é vazia, false, caso contrário.

verificarFilaCheia - devolve true se não há mais espaço.

Manipulação:

pushFila – acrescenta um novo elemento no final da fila (após o último).

popFila – remove o primeiro elemento da fila.

esvaziarFila – torna a fila vazia.

Vamos considerar uma fila de números inteiros nas declarações a seguir.

### INTERFACE

```
void criarFilaVazia(FilaCircular *);
int acessarInicio(FilaCircular *);
bool verificarFilaVazia(FilaCircular *);
bool verificarFilaCheia(FilaCircular *);
void push(FilaCircular *, int);
void pop(FilaCircular *);
void esvaziarFila(FilaCircular *);
```

### IMPLEMENTAÇÃO 1 - Estrutura de armazenamento de dados em ARRAY – Fila circular

```
# define MaxFila 8          /* a fila pode guardar até 8 elementos */
typedef struct{
    int inicio;              /* inicio indica o local do primeiro da fila */
    int fim;                 /* fim indica o local do ultimo da fila */
    int item [MaxFila];     /* o array item guarda os elementos a partir da posição zero */
}FilaCircular;
```

A fila representada ao lado é: 15, 17, 51, 11, 15, 37. O início da fila é na posição 4 e o fim da fila na posição 1.

O modelo dessa representação é uma estrutura sequencial disposta em círculo. O próximo item colocado na fila ocupará a posição 2, após o último. A próxima retirada da fila deverá alterar o apontador inicio para 5.

0	15	
1	37	inicio
2	10	
3	25	
4	15	
5	17	
6	51	
7	11	

4

fim

1

### EXERCICIOS

- 1) IMPLEMENTAR o tipo FilaCircular declarado acima, considerando uma fila de números inteiros positivos e a estrutura de dados usando ARRAY.
- 2) IMPLEMENTAR o tipo Fila declarado a seguir:

```
# define MaxFila 8          // a fila pode guardar até 8 elementos
typedef struct{
    int inicio;              // inicio indica o local do primeiro da fila
    int fim;                 // fim indica o local do ultimo da fila
    int item [MaxFila];     // o array item guarda os elementos a partir da posição zero
} c_Fila;
typedef c_Fila * a_Fila;

// interface
a_Fila criarFilaVazia( );    // o construtor cria a fila vazia
int acessarFila(a_Fila);     // devolve o primeiro da fila; devolve -1 caso a fila seja vazia
bool verificarFilaVazia(a_Fila); // devolve TRUE se a fila estiver vazia
bool verificarFilaCheia(a_Fila); // devolve TRUE se não há mais lugar na fila
void pushFila(a_Fila, int);  // coloca o valor dado no fim da fila
void popFila(a_Fila);        // retira o primeiro da fila
void esvaziarFila(a_Fila);   // esvazia a fila, deixando-a vazia
```