3. Fila Circular em C - Estrutura de Dados

3.1 Conceito

- O conceito é semelhante ao de uma FILA qualquer, no entanto uma **fila circular** tem o objetivo de reaproveitamento total dos espaços, que por ventura, deixarão de ter novas utilizações e eliminando toda e qualquer possibilidade de deslocamentos na estrutura de armazenamento;
- **FIFO** ("First In, First Out"), ou seja, o primeiro a entrar será o primeiro a sair;
- Todo e qualquer acesso deve ocorrer, exclusivamente, pelos *controladores* da fila, sendo:
  - Início -> somente para Retiradas;
  - Fim -> somente para Inserções.

3.2 Exemplos de Aplicação

- Pode ser utilizado os mesmo exemplos de uma fila qualquer, só que com o reaproveitamento de espaços;
- Troca de mensagens entre computadores num rede;
- Controle de documentos para impressão;
- Filas de banco (sem prioridade);
- Filas de supermercado;
- Um semáforo;
- Entre outros.

3.3 Estrutura da Fila Circular

A Estrutura de Dados e algumas operações precisam ser repensadas para que a implementação das mesmas reflita a otimização de acessos e utilização dos espaços disponíveis.

TAD (Tipo Abstrato de Dados) da Fila Circular:

```
#define MAXFILA 10

struct TpFilaCirc{
    int inicio, fim, cont;
    char fila[MAXFILA];
};
```

O controle de quantidade (cont) de elementos, facilita identificar se a Fila Circular está Cheia ou Vazia.

3.4 Operações Associadas

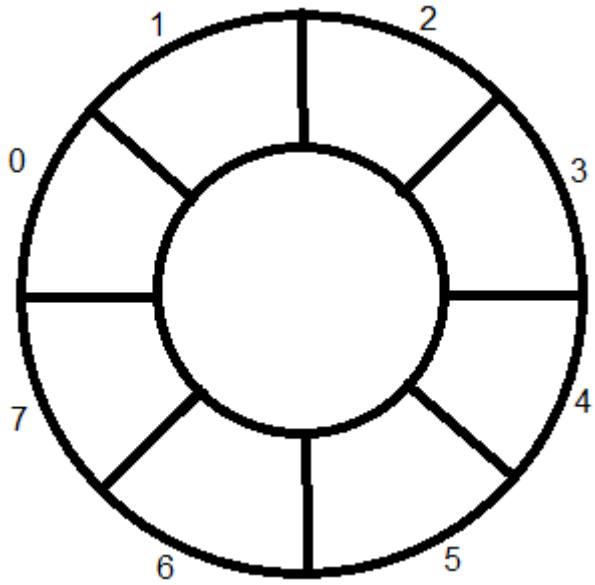
- void FCInicializar (TpFilaCirc &F)
- void FCInserir (TpFilaCirc &F, char elemento)
- char FCRetirar (TpFilaCirc &F)
- char FCElementoInicio (TpFilaCirc F)
- char FCElementoFIM (TpFilaCirc F)
- char FCVazia (int cont)
- char FCCheia (int cont)
- void FCExibir (TpFilaCirc F)

3.5 Implementação em C

```
#define MAXFILA 10

struct TpFilaCirc{
    int inicio, fim, cont;
    char fila[MAXFILA];
};

void FCInicializar(TpFilaCirc &F){
    F.inicio = 0;
    F.fim = -1;
    F.cont = 0;
}

void FCInserir(TpFilaCirc &F, char elemento){
    if(F.fim == MAXFILA -1)   // se o fim for a ultima posição do vetor, ele
recebe -1 para assim ser inserido na posição 0 (o inicio)
        F.fim = -1;

    F.fila[++F.fim] = elemento;

    F.cont ++;
}
char FCRetirar ( TpFilaCirc &F){
    int aux;

    aux = F.fila[F.inicio++];      // a variável aux recebe o elemento do início
da fila, o que vai ser retirado

    if(F.inicio == MAXFILA)     // se o inicio for == MAXFILA ele recebe 0, ele
"reseta", volta pro início
        F.inicio = 0;

    F.cont --;  //decrementa o contador, ja que foi retirado um elemento

    return aux;
}

char FCElementoInicio(TpFilaCirc F){
```

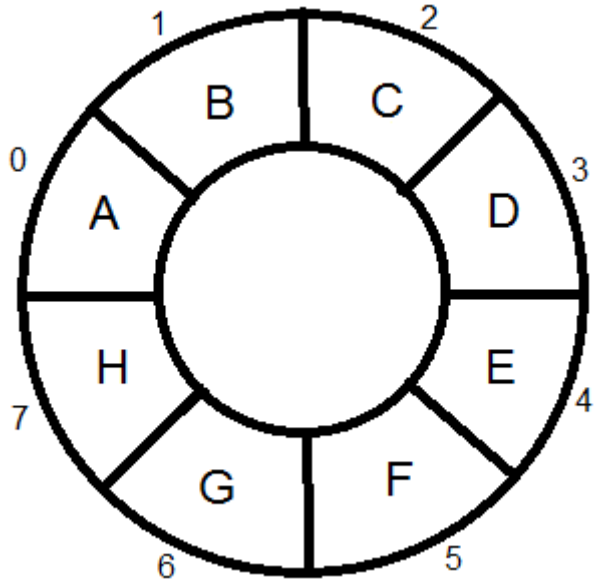
```
    return F.fila[F.inicio];
}
char FCElementoFIM(TpFilaCirc F ){
    return F.fila[F.fim];
}

char FCVazia(int cont){
    return cont == 0;
}
char FCCheia(int cont){
    return cont == MAXFILA;
}

void FCExibir(TpFilaCirc F){
    while(!FCVazia(F.cont))
        printf("\t%c", FCRetirar(F));
}
```

3.6 Funcionamento

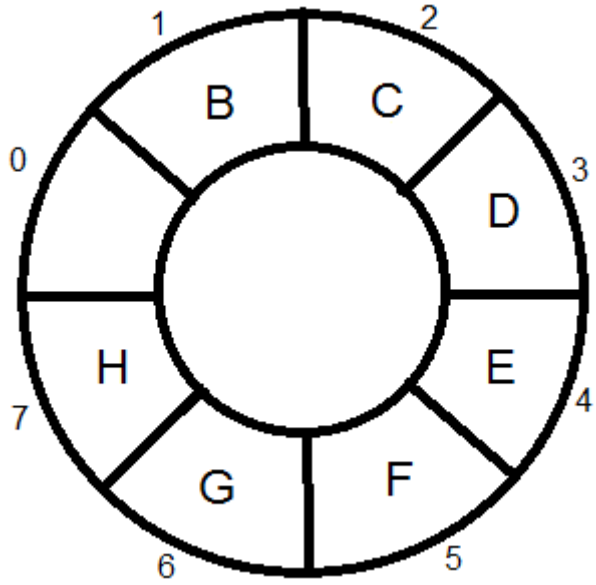
1. Inicializar:
    - Início = 0
    - Fim = -1
    - Contador = 0
    - Fila = 

2. Inserir 'A' / Inserir 'B' / Inserir 'C' / Inserir 'D' / Inserir 'E' / Inserir 'F' / Inserir 'G' / Inserir 'H':
    - Início = 0
    - Fim = 7
    - Contador = 8

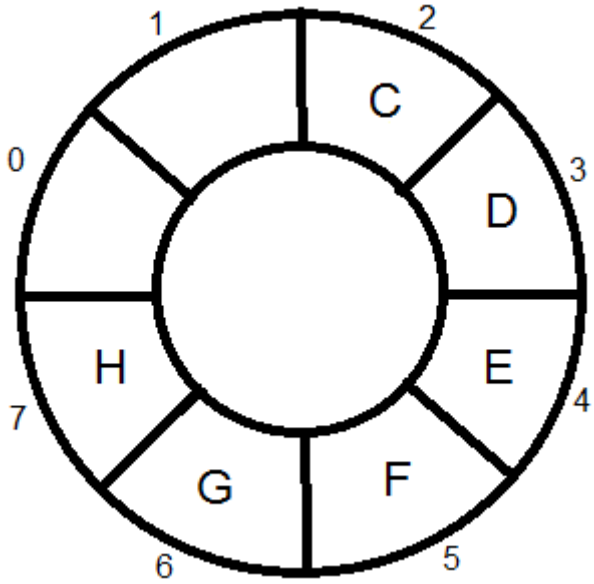
- Fila =

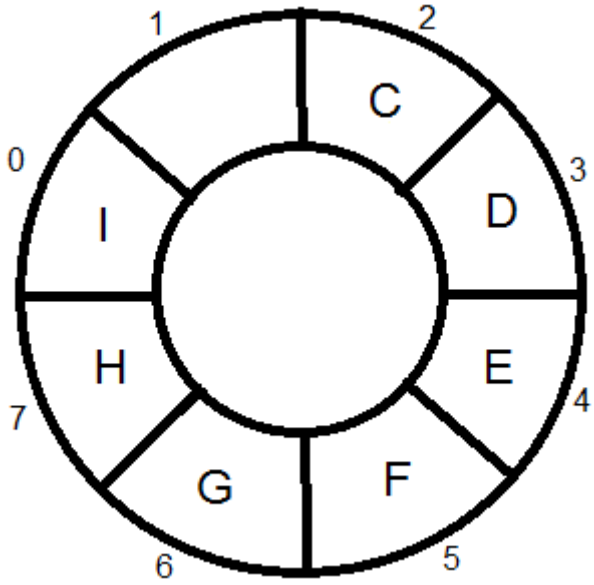


3. Retirar (obrigatoriamente retorna o elemento do início da fila):
    - Início = 1
    - Fim = 7
    - Contador = 7
    - Fila =



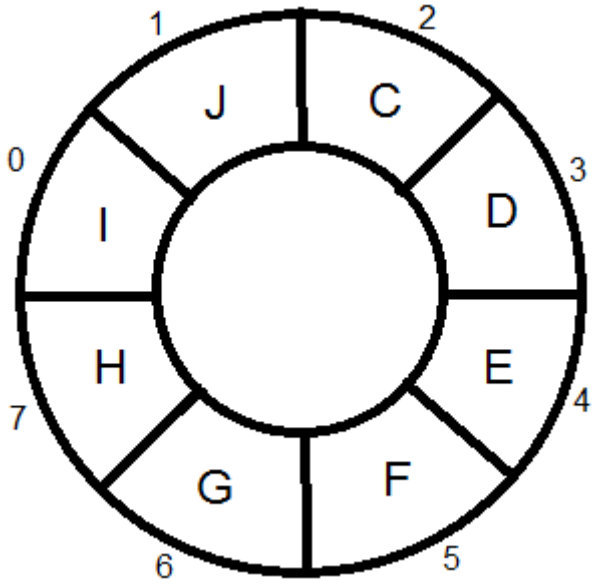
4. Retirar:
    - Início = 2
    - Fim = 7
    - Contador = 6

- Fila = 

5. Inserir 'I':
    - Início = 2
    - Fim = 0
    - Contador = 7
    - Fila = 

6. Inserir 'J':
    - Início = 2
    - Fim = 1
    - Contador = 8

- Fila =



7. Retirar:

- Início = 3
- Fim = 1
- Contador = 7
- Fila =