

Alessandro Calin

Fila Circular



ESTRUTURA DE DADOS
alessandrocalin@gmail.com
967617845

2 – 8 – 4 – 5 – 1 – 6 – 10 – 3 – 9 – 12

Fila - FIFO



$p \rightarrow R$ $p \rightarrow F$

Struct com:

Vetor

F (frente) - retirar

R (retaguarda) – Inserir

Para programar teríamos um
ponteiro (p)



Qual era a problemática da fila?

Temos duas variáveis para controlar a inserção e a retirada (R e F). Quando a variável R chega ao seu limite, significa que fila está cheia. Quando a variável F chega ao seu limite, significa que a fila está vazia. Neste contexto R e F passam a ficar juntas no topo da fila. Com isso não conseguimos alocar novos elementos na fila provocando um desperdício de memória.

Com este contexto formado passamos a ter as filas circulares. A maior diferença é que teremos uma nova variável auxiliar para voltar a alocar elementos evitando desperdiçar memória



Passo a passo para implementar uma fila circular

- 1) Importar as Bibliotecas;**
- 2) Definir a Struct (4 elementos);**
- 3) Criar a função enfileirar;**
 - 3.1) Definir a função enfileirar;**
 - 3.2) Verificar se a fila está cheia. A fila circular está cheia quando a variável “Q” for igual a TAM;**
 - 3.3) Verificar se a variável R já chegou no limite (TAM-1), caso tenha chegado no limite voltar com ela para o primeiro índice da fila (zero). Caso ao contrário a variável R será incrementada.**
 - 3.4) Armazenar o valor dentro da fila;**
 - 3.5) Incrementar a variável Q**



```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#define TAM 10
```

```
typedef struct{  
    int R, F, Q;  
    int vet[TAM];  
}filacircular;
```

```
void enfileirar(int valor, filacircular *p){  
    if (p -> Q == TAM+1){  
        printf("\n Fila Cheia!");  
        exit(0);  
    }  
    if (p -> R == TAM-1){  
        p -> R = 0;  
    }  
    else{  
        p -> R++;  
    }  
    p -> vet[p -> R] = valor;  
    p -> Q++;  
}
```

```
int desenfileirar (filacircular *p){  
    int aux;  
    if (p -> Q == 0){  
        printf("\nFila Vazia!");  
        exit(1);  
    }  
    aux = p -> vet[p -> F];  
    if ( p -> F == TAM-1){  
        p -> F = 0;  
    }  
    else {  
        p -> F++;  
    }  
    p -> Q --;  
    return aux;  
}
```

```
int main(void){
    filacircular fila;
    int valor;
    fila.R = -1;
    fila.F = 0;
    fila.Q=0;
    for(int i=0; i< TAM; i++){
        printf("Digite o valor a ser enfileirado: \n");
        scanf("%d", &valor);
        enfileirar(valor, &fila);
    }
    for(int i=0; i < TAM; i++){
        printf("\n%d", desenfileirar(&fila));
    }
    system("pause");
}
```