



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 6

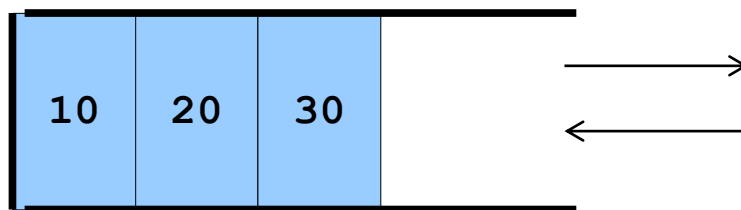
Estrutura: FILAS

Representadas por Contiguidade Física
FilaCF

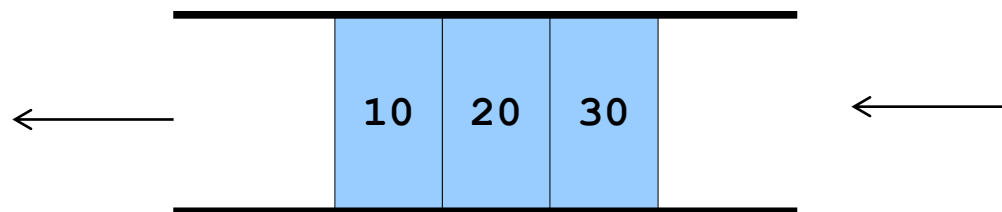
Listas Lineares com Disciplina de Acesso

São listas lineares onde o acesso a seus nodos é realizado nas extremidades.

- Pilha (Stack)



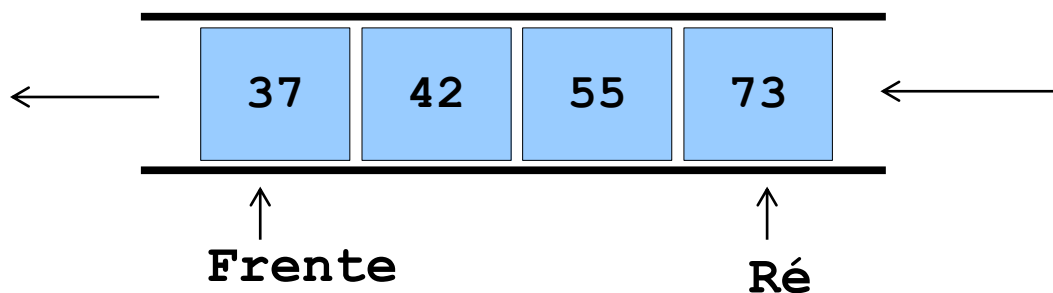
- Fila (Queue)



FILA – representada por Contiguidade física de memória

É uma lista onde a **inserção** é realizada em uma extremidade e a **retirada** na outra.

Estrutura Lógica:



Critério "FIFO" (First In First Out)
(primeiro que entra é o primeiro que sai)

Estrutura Física:

Endereços		
1001	37	nodo 1
1002		
1003	42	nodo 2
1004		
1005	55	nodo 3
1006		
1007	73	nodo 4
1008		
1009		nodo 5
1010		
1011	0	frente
1012		
1013	3	ré

FILA: por Contiguidade Física

```
// códigos de retorno
#define SUCESSO 0
#define FILA_VAZIA 1
#define FILA_CHEIA 2

// máximo de elementos
#define MAX_NODOS 5.

typedef struct {
    int v[MAX_NODOS];
    int frente;
    int re;
} Filacf;
```

Operações:

```
void criaFila (FilaCF *f);
int insere (FilaCF *f,int dado);
int retira(FilaCF *f,int *dado);
int estaCheia(FilaCF f);
int estaVazia(FilaCF f);
int consulta(FilaCF f,int *dado);
```

Inicializar uma FILA (criaFila)

Função: criaFila

Saída: uma fila vazia

Retorno: nenhum

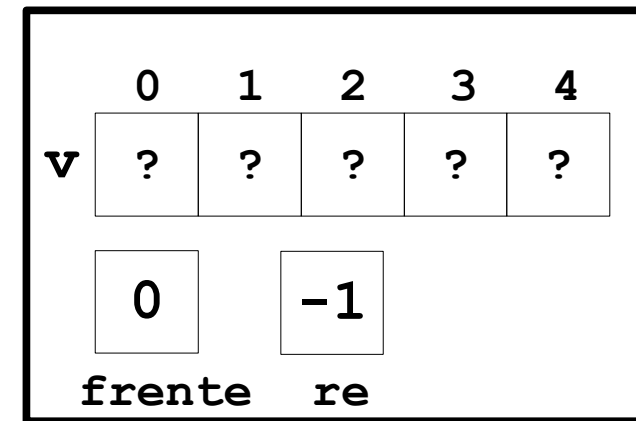
Descrição: Cria uma fila vazia

```
void criaFila(FilaCF *pl);
```

```
int main() {  
    FilaCF fl;  
    criaFila(&fl);  
    ...  
}
```

```
void criaFila(FilaCF *fl) {  
    fl->frente = 0;  
    fl->re = -1;  
}
```

fila vazia



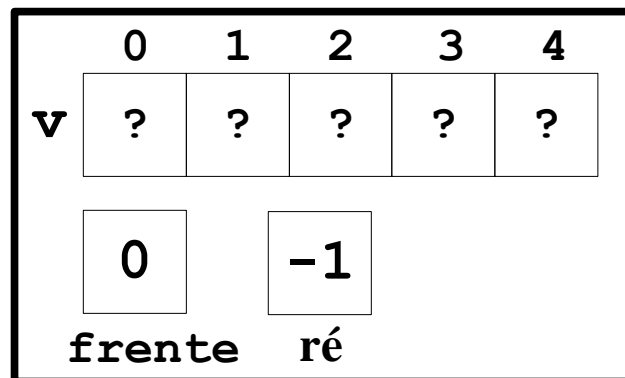
v : Vetor que armazena os nodos da pilha.

frente : índice do nodo que está no início da fila.

re: índice do nodo que está no fim da fila.

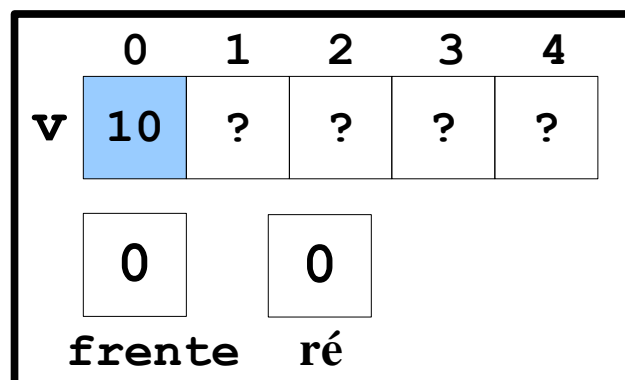
FILA – Insere

```
int main() {  
    FilaCF f;  
    criaFila(&f);  
}
```

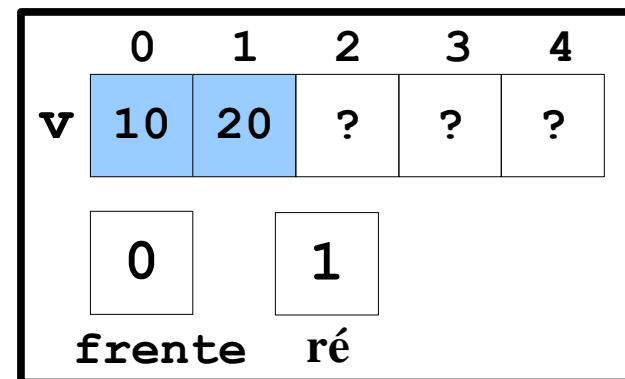


```
// Na inclusão a ré é incrementada  
// re++;
```

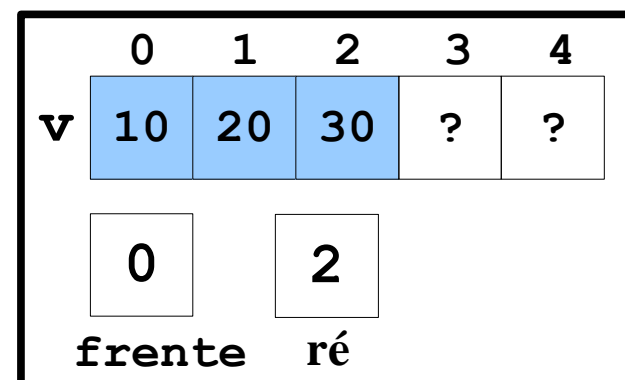
```
insere(&f, 10);
```



```
insere(&f, 20);
```



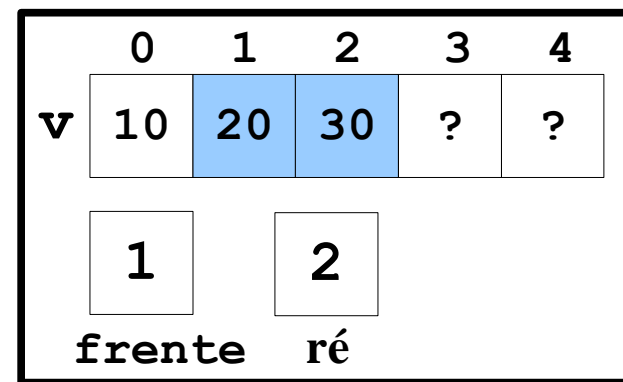
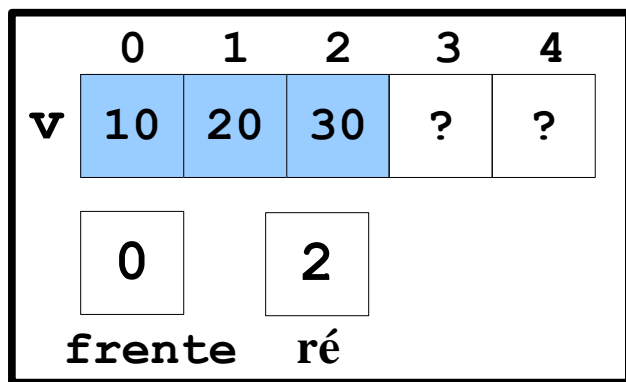
```
insere(&f, 30);
```



FILA – Retira

```
// Na exclusão a frente é incrementada  
// frente++;
```

```
retira(&f, &x);
```



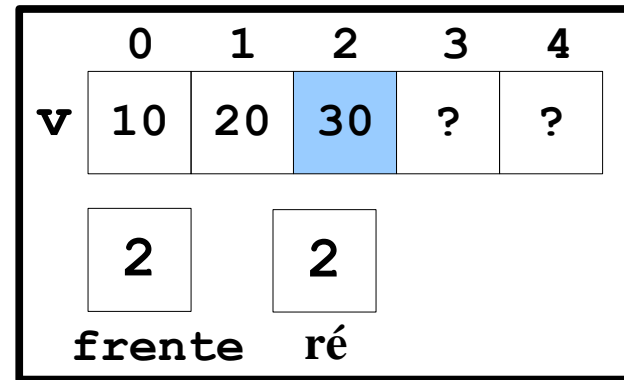
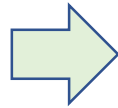
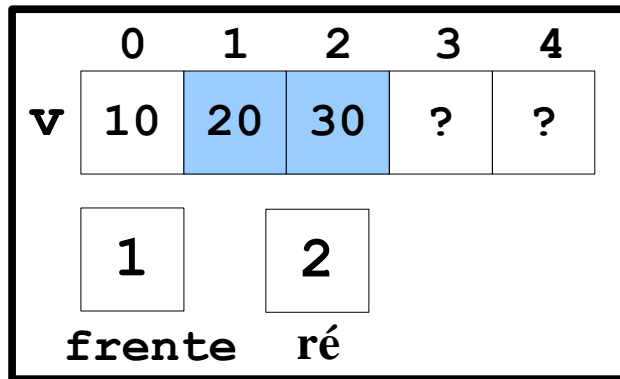
Como verificar se a fila está vazia?
Como verificar se a fila está cheia?

FILA – Fila vazia?

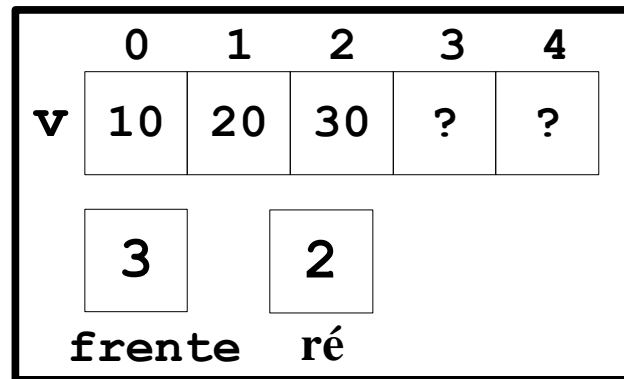
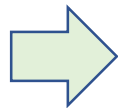
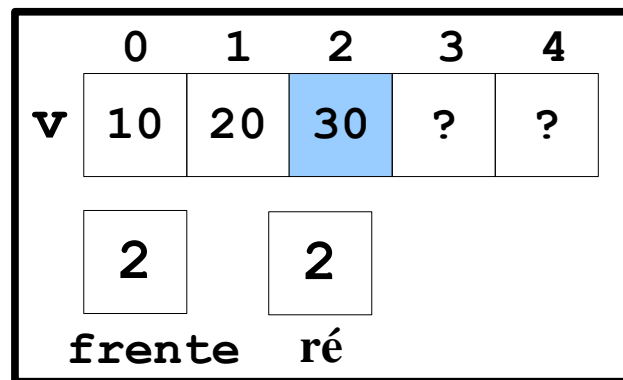
Como verificar se a fila está vazia?

Ré < Frente

`retira(&f, &x);`



`retira(&f, &x);`

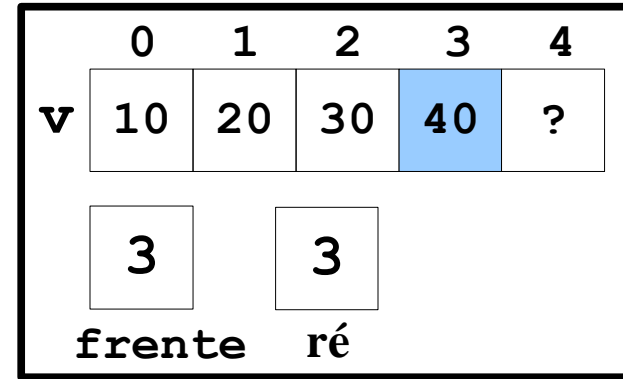
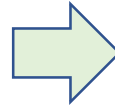
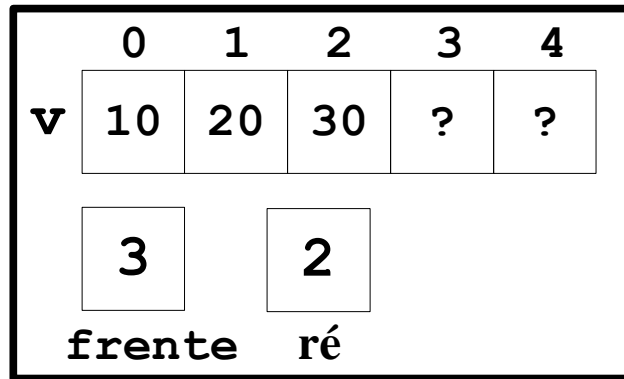


FILA – Fila cheia?

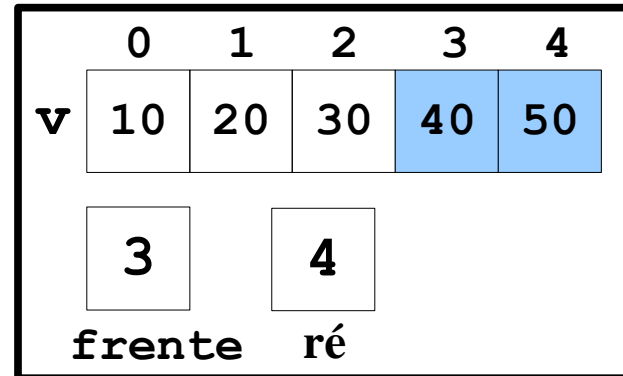
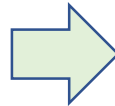
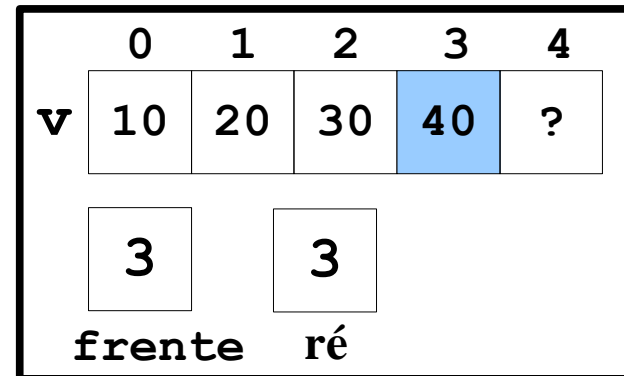
Como verificar se a fila está cheia?

$$R\acute{e} = MAX_NODOS - 1$$

`insere(&f, 40);`



`insere(&f, 50);`



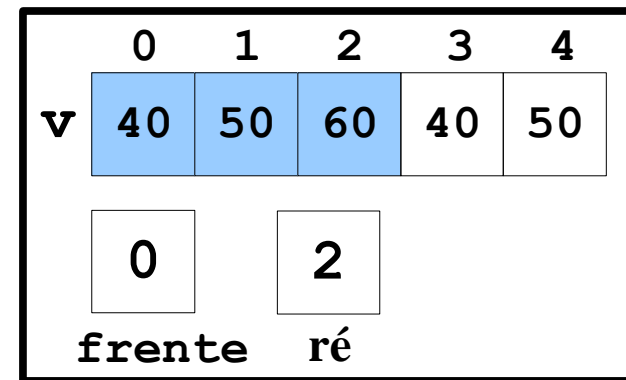
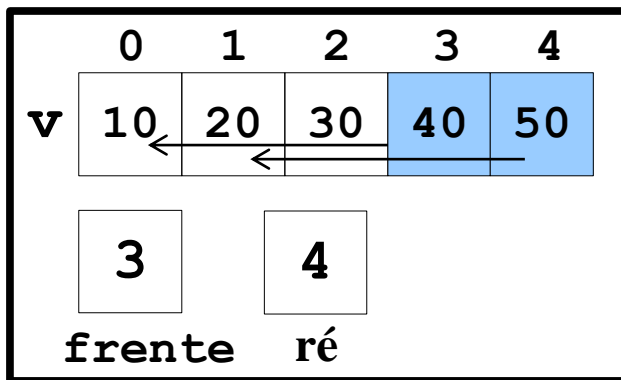
FILA – com reaproveitamento de espaço

Na forma básica da FILA **não há** reaproveitamento do espaço desocupado.

Como reaproveitar o espaço?

1) Deslocando os nodos para o início do vetor **na situação de fila cheia**.

`insere (&f, 60) ;`

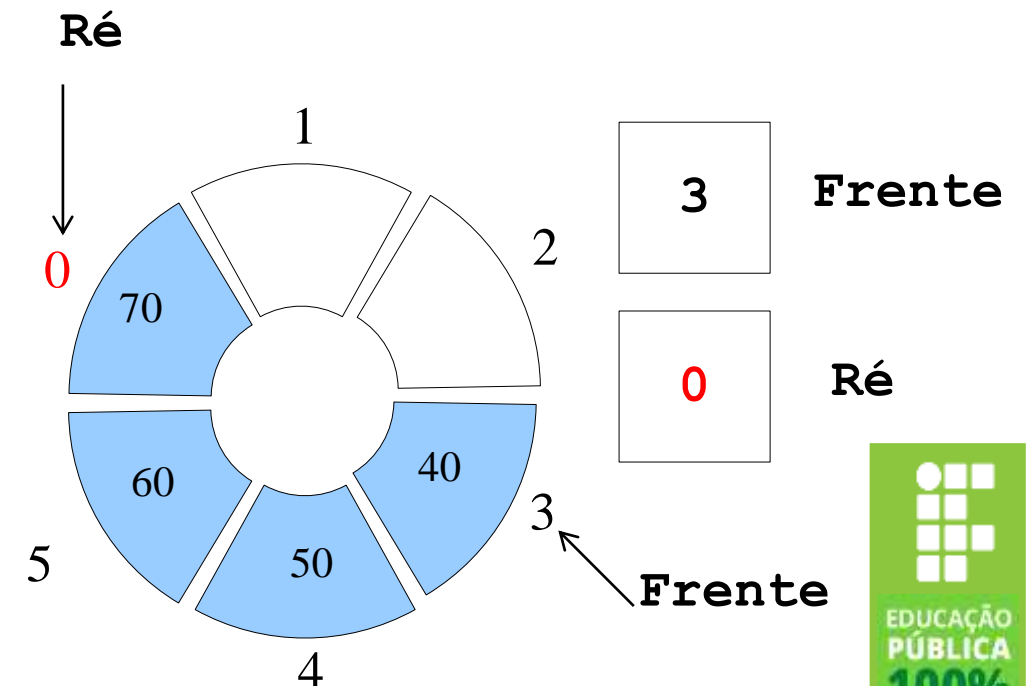
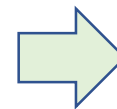
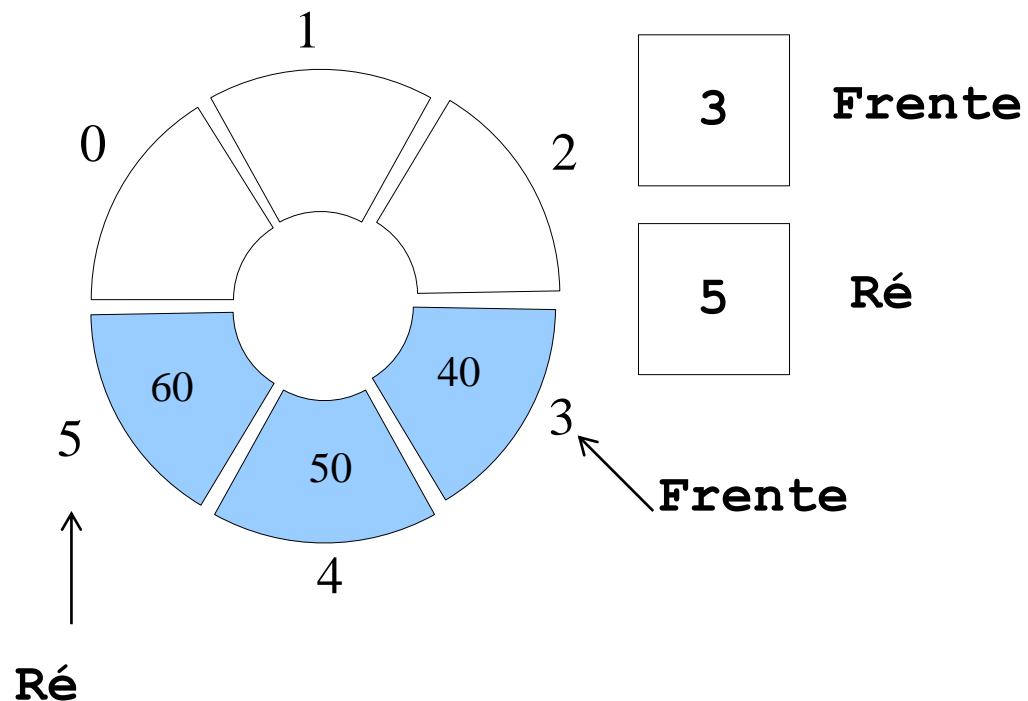


FILA Circular

Como reaproveitar o espaço?

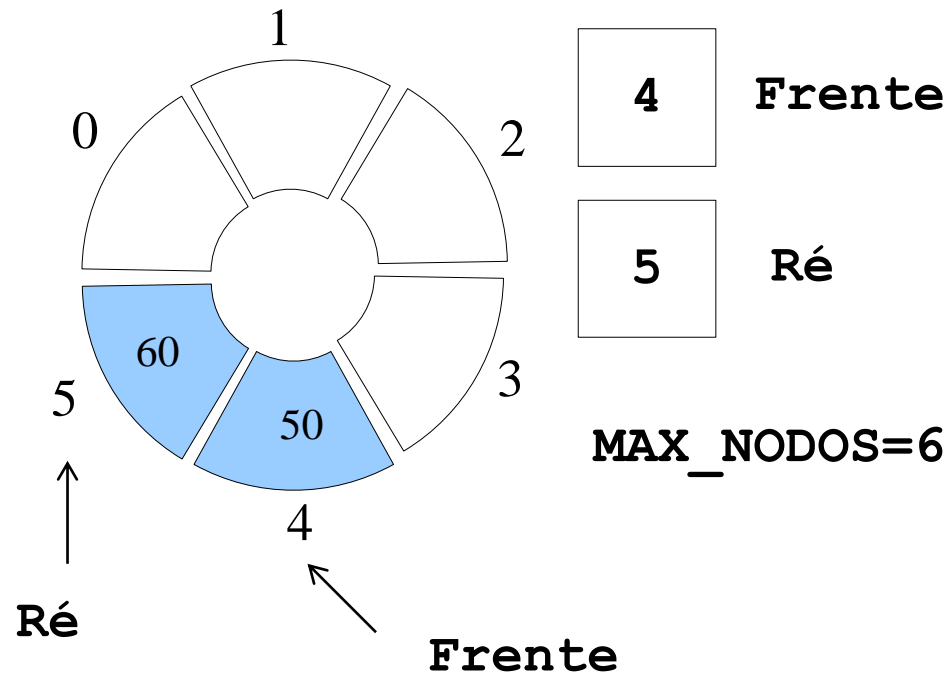
2) Criando uma **fila circular**, onde o último elemento é considerado adjacente ao primeiro.

```
insere (&f, 70) ;
```



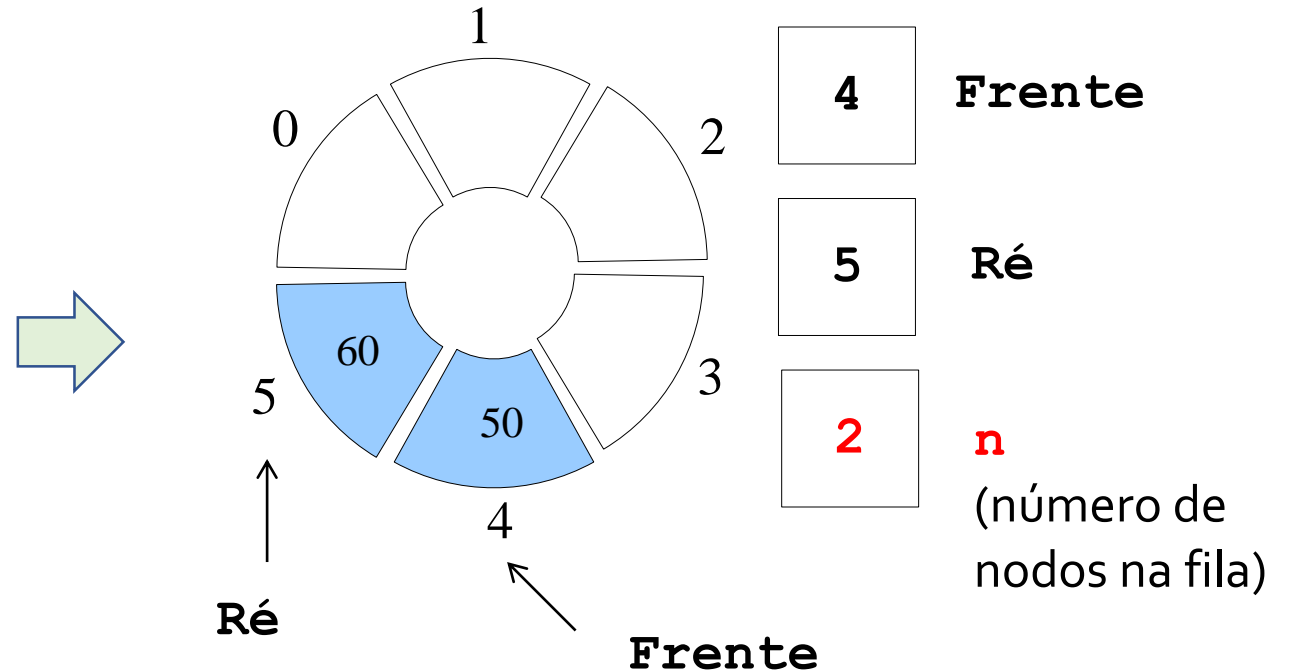
FILA Circular

Como verificar se a fila está cheia?



Ré = MAX_NODOS - 1
e a fila **não** está cheia

Solução: Incluir um controle de tamanho:



Como verificar se a fila está vazia? **n=0**
Como verificar se a fila está cheia? **n=MAX_NODOS**

FILA: por Contiguidade Física

```
// códigos de retorno
#define SUCESSO 0
#define FILA_VAZIA 1
#define FILA_CHEIA 2

// máximo de elementos
#define MAX_NODOS 5.

typedef struct {
    int v[MAX_NODOS];
    int frente;
    int re;
} Filacf;
```

Operações:

```
void criaFila (FilaCF *f);
int insere (FilaCF *f,int dado);
int retira(FilaCF *f,int *dado);
int estaCheia(FilaCF f);
int estaVazia(FilaCF f);
int consulta(FilaCF f,int *dado);
```



INSTITUTO FEDERAL
Sul-rio-grandense

Câmpus
Pelotas

EDUCAÇÃO
PÚBLICA
100%
GRATUITA

Estrutura de Dados

Aula 6

Estrutura: FILAS

Representadas por Contiguidade Física
FilaCF