

Algoritmos e Estruturas de Dados III

FILA CIRCULAR

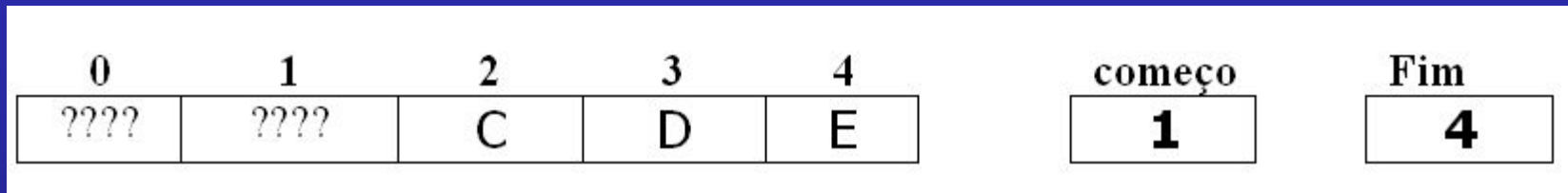
FILA SEQUENCIAL

O que ocorre se tentarmos inserir mais um elemento na fila abaixo ?

0	1	2	3	4	começo	Fim
????	????	C	D	E	1	4

FILA SEQUENCIAL

O que ocorre se tentarmos inserir mais um elemento na fila abaixo ?



Note que **TAM** = 5 e que portanto a última posição física que pode ser ocupada é a posição 4

```
boolean cheia () {  
    if (FIM == TAM - 1) {  
        return true;  
    }  
    return false;  
}
```

FILA CIRCULAR

APESAR DE EXISTIREM “LOCAIS” DESOCUPADOS,

A FILA SEQUENCIAL NÃO PERMITE QUE SEJAM REAPROVEITADOS.

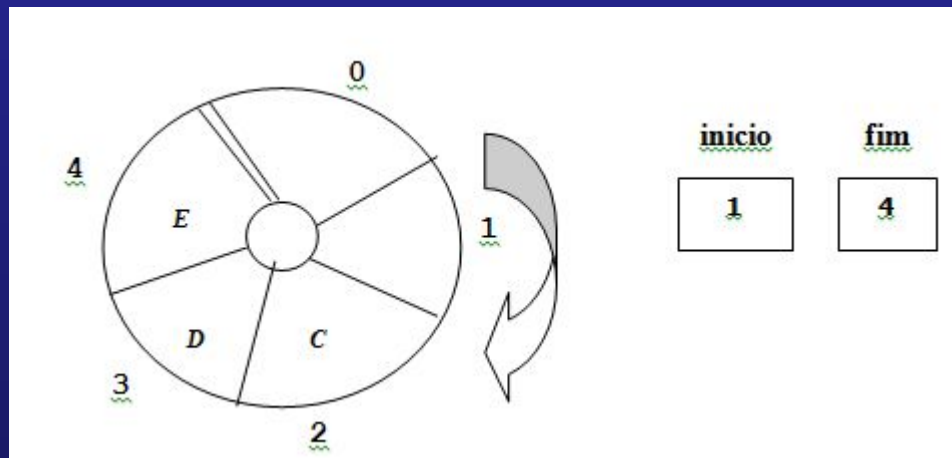
A ESTRUTURA DE DADOS
QUE PERMITE ESSE REAPROVEITAMENTO
É A **FILA CIRCULAR**.

FILA CIRCULAR

PRINCÍPIO BÁSICO:

Para calcular a próxima posição, usar:
 $(FIM + 1) \% CAPACIDADE$ ao invés de **FIM++**.

Isto permite que posição 0 do vetor seja vista como sendo uma EXTENSÃO da ÚLTIMA posição e assim por diante..



FILA CIRCULAR

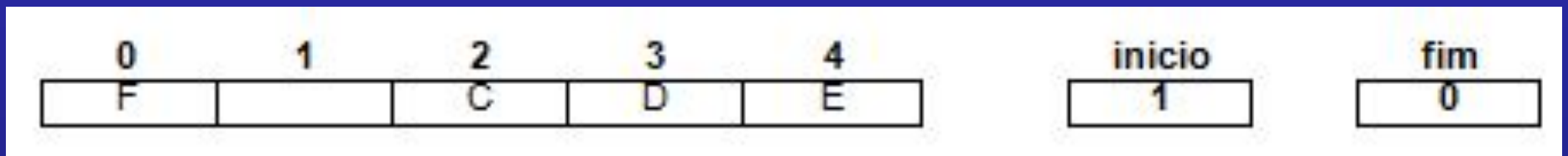
EXEMPLO

Considere capacidade = 5

i	i+1	$(i+1) \% \text{ capacidade}$
0	1	$1 \% 5 = 1$
1	2	$2 \% 5 = 2$
2	3	$3 \% 5 = 3$
3	4	$4 \% 5 = 4$
4	5 INDICE INVÁLIDO !	$5 \% 5 = 0$ INDICE VÁLIDO!

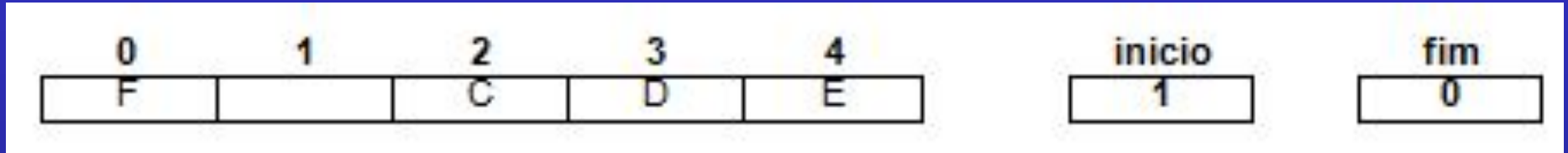
FILA CIRCULAR - INCLUSÃO

Na abaixo, ao se tentar inserir um elemento na mesma (e há espaço para isso pois a posição 1 do vetor está livre), o valor de **inicio** será igual a **fim** (ambos valerão 1) e isto configura **Fila Vazia** !



TORNA-SE NECESSÁRIO REVISAR OS CONCEITOS
DE FILA CHEIA, VAZIA E INCLUSÃO....

FILA CIRCULAR - INCLUSÃO



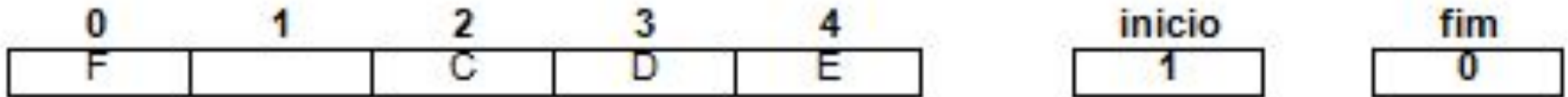
A idéia agora é imaginar o seguinte:

Se ao tentar inserir um novo elemento (**no fim**), atingir a posição **inicio**, significa que a **FILA ESTÁ CHEIA** !

Nesta solução, trabalha-se com o conceito de *HEADER* (cabeçalho) ou *SENTINELA* (guarda), ou seja, “desperdiça-se” uma posição do vetor que será usada como separador entre o **inicio** e o **final** da fila circular.

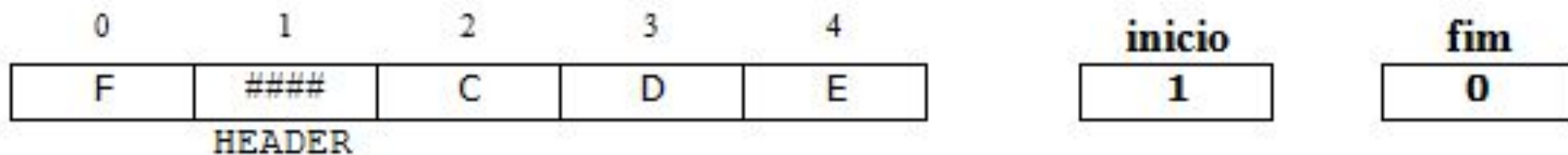
Isto é necessário para se diferenciar a situação de **fila cheia** e **vazia**.

FILA CIRCULAR - INCLUSÃO



Nesta solução, trabalha-se com o conceito de **HEADER** (cabeçalho) ou **SENTINELA** (guarda), ou seja, “desperdiça-se” uma posição do vetor que será usada como separador entre o **inicio** e o **final** da fila circular.

Isto é necessário para se diferenciar a situação de **fila cheia** e **vazia**.



FILA CIRCULAR – FILA CHEIA

Como detectar se uma Fila Circular está CHEIA ?

Não dá mais para dizer que a fila está cheia se **FIM == CAPACIDADE -1**

Note que abaixo temos uma situação de FILA CHEIA, pois:

0	1	2	3	...	9		começo		Fim
###	João 65	Bia 13	Luiza 32	Juraci 54	Camilo 35		0		9

```
if ( (fim +1) % capacidade == começo ) {  
    system.out.println (" Erro : A Fila esta cheia\n");  
    result = 'N' ;  
}
```

Classe TADFilaCircular

```
class TADFilaCircular {  
    private int inicio, fim;  
    private int capacidade;  
    private TPessoa [] pessoa;
```

```
// Construtor
```

```
public TADFilaCircular (int tam) {  
    capacidade = tam;  
    inicio = 0;    // O header começa na posição ZERO  
    fim     = 0;  
    pessoa = new TPessoa [capacidade];  
}
```

Classe TADFilaCircular

// Métodos Auxiliares

```
public boolean filaCheia () {
    if (inicio == (fim + 1) % capacidade) return true;
    return false;
}

public boolean filaVazia () {
    if (inicio == fim ) return true;
    return false;
}

public int contaFila( )      {

}

}
```

A CLASSE FILA CIRCULAR

```
public boolean enqueue (TPessoa aux ) {  
    if (filaCheia()) return false;  
  
    fim = (fim + 1) % capacidade;  
    pessoa [fim] = new TPessoa (aux);  
    return true;  
}
```

A CLASSE FILA CIRCULAR

```
public TPessoa desenfileira () {  
    if (filaVazia()) return null;  
  
    TPessoa aux = pessoa [inicio];  
    inicio      = (inicio + 1) % capacidade;  
    return aux;  
}
```

A CLASSE FILA CIRCULAR

```
public void imprimeFila () {  
    if (filaVazia()) System.out.println("Fila Vazia!");  
    else {  
        int posi;  
        for (int i=1; i <= contaFila(); i++) {  
            posi = (inicio+i) % capacidade;  
            System.out.println ( pessoa[posi].nome + pessoa[posi].idade);  
        }  
    }  
}
```