

FILAS

Prof.: Julio Cesar dos Reis

Roteiro

2

- Conceito de fila
- Implementação com arranjos
- Implementação com estruturas ligadas

Filas (Queue)

3

- Primeiro a entrar é o primeiro a sair
 - ▣ Respeita-se a ordem de chegada dos elementos
- Remove-se primeiramente os elementos inseridos há **MAIS** tempo

Filas (Queue)

4

- Primeiro a entrar é o primeiro a sair
 - ▣ Respeita-se a ordem de chegada dos elementos
- Remove-se primeiramente os elementos inseridos há **MAIS** tempo
- Inserção e remoção se faz por extremidades distintas da estrutura
 - ▣ Inserção => final da sequência
 - ▣ Remoção => início da sequência



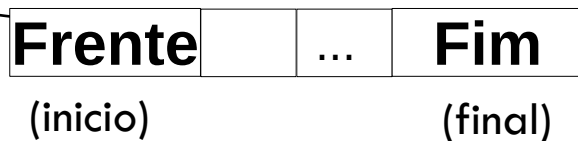
TAD Fila

5

□ Operações

Desenfileirar
(remove)

Enfileirar
(incluir)

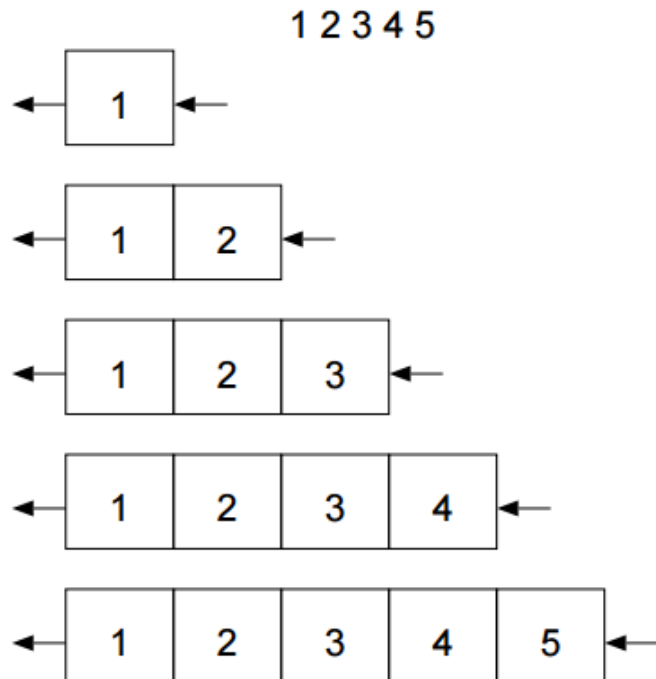


- enfileirar
- desenfileirar
- filaVazia
- filaCheia
- tamanho

Exemplo de operações em fila

6

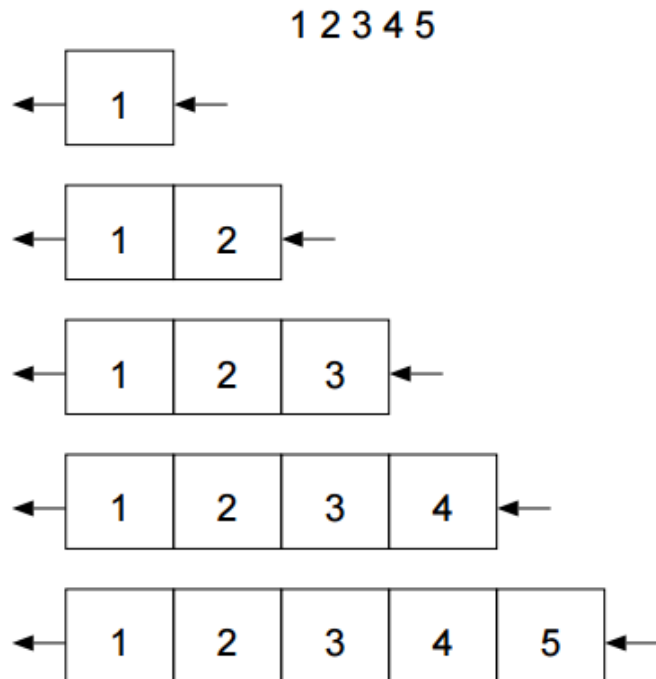
□ Sequência de enfileiramento



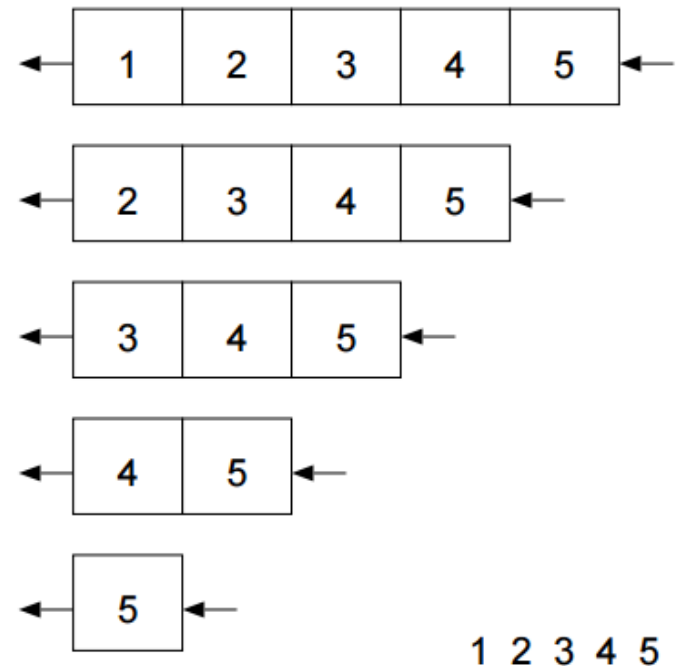
Exemplo de operações em fila

7

□ Sequência de enfileiramento



□ Sequência de desenfileiramento



Abordagens de implementação

8

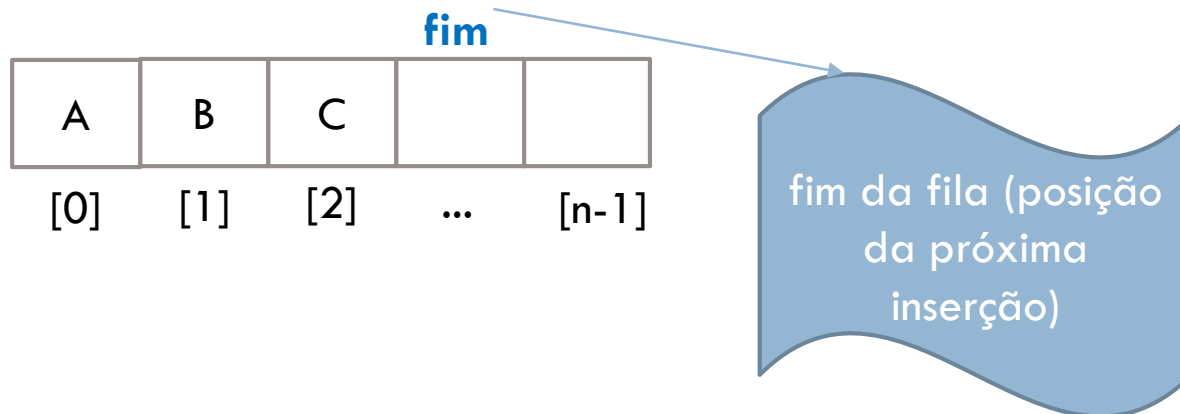
- Sequencial com uso de arranjos
 - ▣ Sequencial circular (fila circular)

- Encadeada usando lista ligada
 - ▣ Lista ligada circular (fila circular)

Implementação sequencial de fila

9

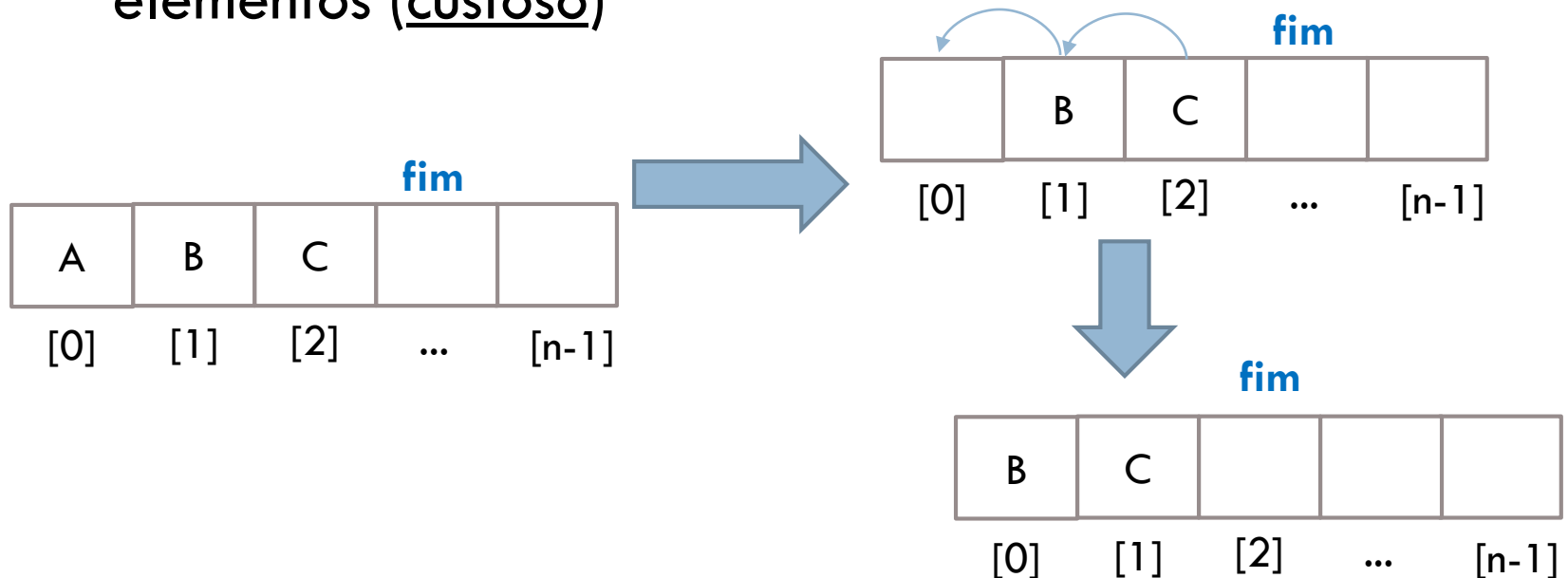
- Uso de vetor (1ª opção)
 - ▣ Tamanho máximo do vetor (n elementos)
 - ▣ Enfileirar: inserir no final do vetor (variável fim)



Implementação sequencial de fila

10

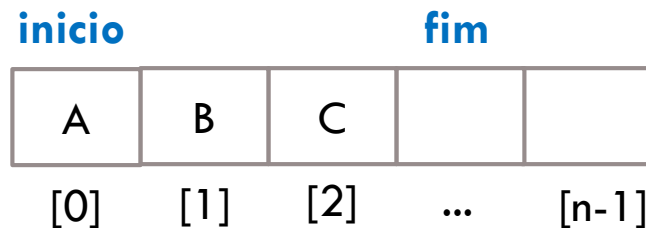
- Uso de vetor (1ª opção)
 - ▣ Tamanho máximo do vetor (n elementos)
 - ▣ Enfileirar: inseri no final do vetor (variável fim)
 - ▣ Desenfileirar: remove no começo do vetor e desloca elementos (custoso)



Implementação sequencial de fila

11

- Uso de vetor (2ª opção)
 - ▣ Tamanho máximo do vetor (n elementos)
 - ▣ Índice de **início** e **fim** da fila
 - ▣ Variável que controla a quantidade de elementos



Estrutura da fila sequencial

12

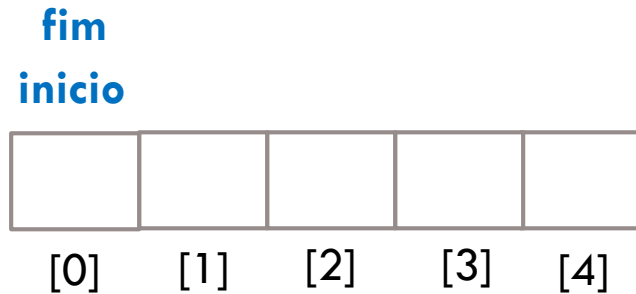
```
typedef struct Fila {  
    int *elems; //alocação de vetor para armazenar os dados  
    //int elems[T_MAX] [#define T_MAX 50;]  
    int inicio; //índice de inicio  
    int fim; //índice de fim  
    int qtd; //quantidade de elementos registrados  
} Fila;
```

Exemplo de Fila (vetor)

13

□ Fila de tamanho máximo igual a 5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



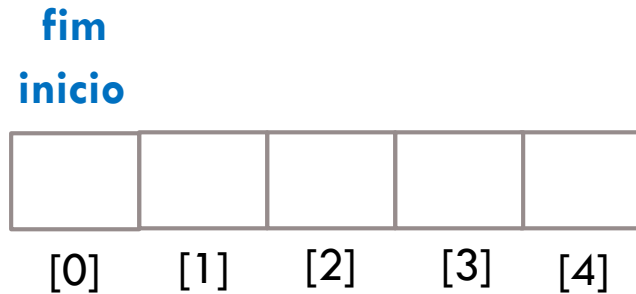
Inicio=0
Fim=0
Qtd=0

Exemplo de Fila (vetor)

14

□ Fila de tamanho máximo igual a 5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=0
Fim=0
Qtd=0

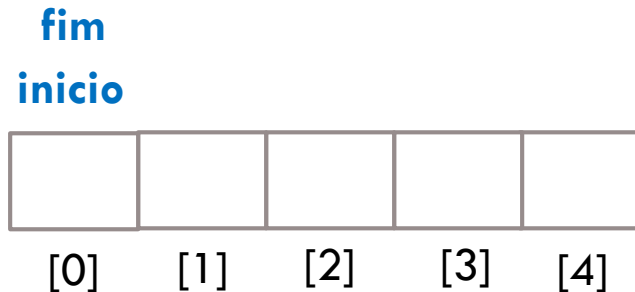
FILA VAZIA
(INICIO==FIM)

Exemplo de Fila (vetor)

15

□ Inicialização

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



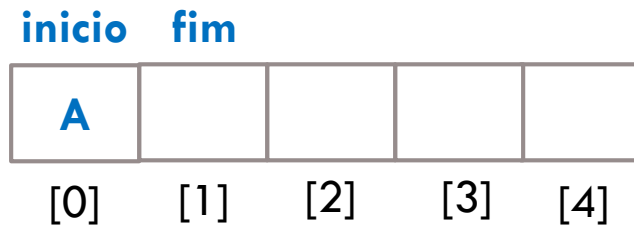
Inicio=0
Fim=0
Qtd=0

```
Fila* inicializarFila(int tam){  
    Fila *f = malloc(sizeof(Fila));  
    f->elems=malloc(sizeof(int*tam));  
    f->inicio=0; f->fim=0; f->qtd=0;  
    return f;  
}
```

Exemplo de Fila (vetor)

16

- **enfileirar(A)**
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)

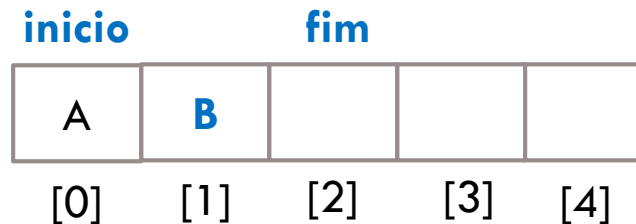


Inicio=0
Fim=1
Qtd=1

Exemplo de Fila (vetor)

17

- enfileirar(A)
- **enfileirar(B)**
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)

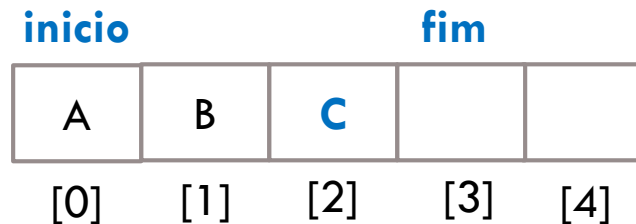


Inicio=0
Fim=2
Qtd=2

Exemplo de Fila (vetor)

18

- enfileirar(A)
- enfileirar(B)
- **enfileirar(C)**
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



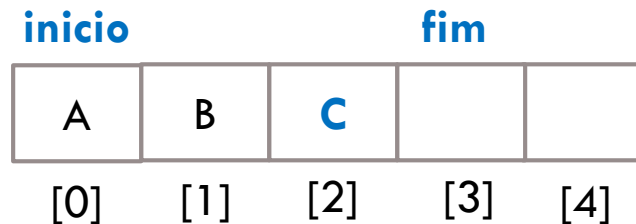
Inicio=0
Fim=3
Qtd=3

```
bool enfileirar(Fila *f, int valor){  
    f->elems[p->fim] = valor;  
    (f->fim)++;  
    return true;  
}
```

Exemplo de Fila (vetor)

19

- enfileirar(A)
- enfileirar(B)
- **enfileirar(C)**
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=0
Fim=3
Qtd=3

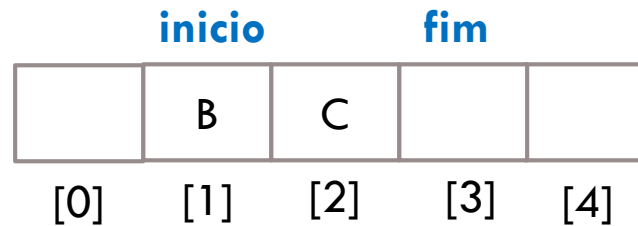
```
bool enfileirar(Fila *f, int valor){  
    f->elems[p->fim] = valor;  
    (f->fim)++;  
    return true;  
}
```

Fila
cheia?

Exemplo de Fila (vetor)

20

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- **desenfileirar()**
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)

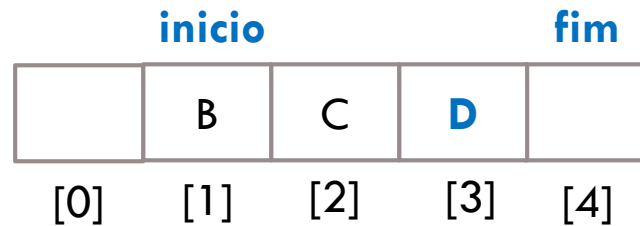


Inicio=1
Fim=3
Qtd=2

Exemplo de Fila (vetor)

21

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- **enfileirar(D)**
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



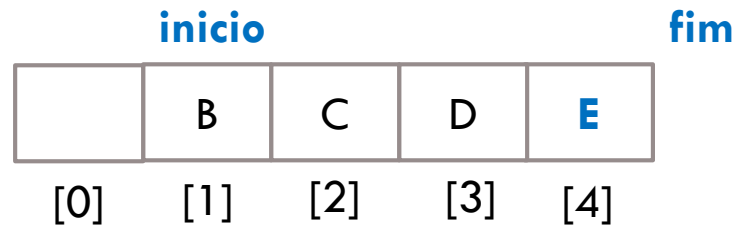
Inicio=1
Fim=4
Qtd=3

QTD=FIM-INICIO

Exemplo de Fila (vetor)

22

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- **enfileirar(E)**
- desenfileirar()
- enfileirar(F)
- enfileirar(G)

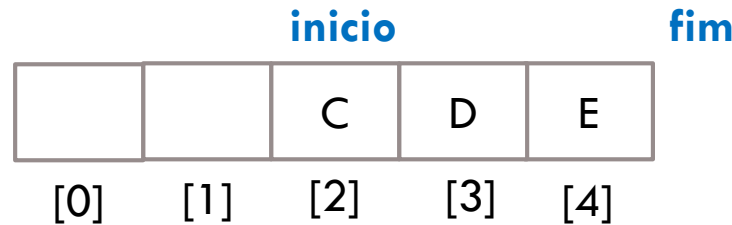


Inicio=1
Fim=5
Qtd=4

Exemplo de Fila (vetor)

23

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- **desenfileirar()**
- enfileirar(F)
- enfileirar(G)



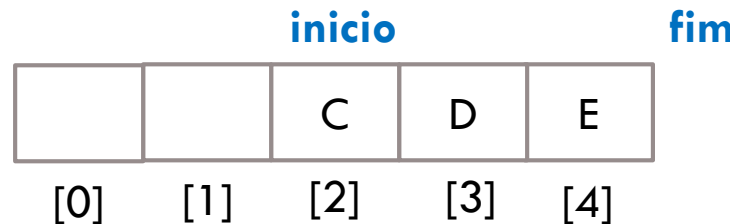
Inicio=2
Fim=5
Qtd=3

```
int desenfileirar(Fila *f){  
    int valor = f->elems[f->inicio];  
    (f->inicio)++;  
    return valor;  
}
```

Exemplo de Fila (vetor)

24

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- **desenfileirar()**
- enfileirar(F)
- enfileirar(G)



Inicio=2
Fim=5
Qtd=3

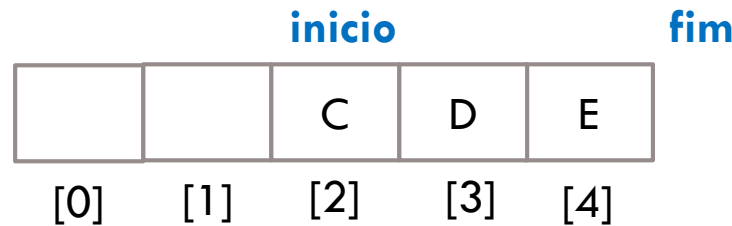
```
int desenfileirar(Fila *f){  
    int valor = f->elems[f->inicio];  
    (f->inicio)++;  
    return valor;  
}
```

Fila
vazia?

Exemplo de Fila (vetor)

25

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- **enfileirar(F)**
- enfileirar(G)

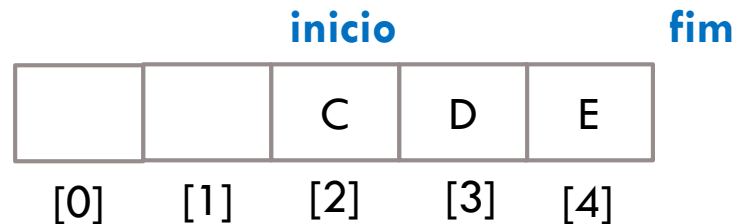


Inicio=2
Fim=5
Qtd=3

Exemplo de Fila (vetor)

26

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- **enfileirar(F)**
- enfileirar(G)



Inicio=2
Fim=5
Qtd=3

□ PROBLEMA

- Tem espaço livre na fila
- Inserção não é permitida
- Mover a fila toda para o inicio do vetor? => custoso

Análise da implementação sequencial

27

- Evita o deslocamento dos elementos
 - ▣ Necessita dos índices *inicio* e *fim* na definição de Fila
- Representação ainda ineficiente
 - ▣ Situação em que a fila está vazia, mas nenhum elemento novo pode ser inserido

Análise da implementação sequencial

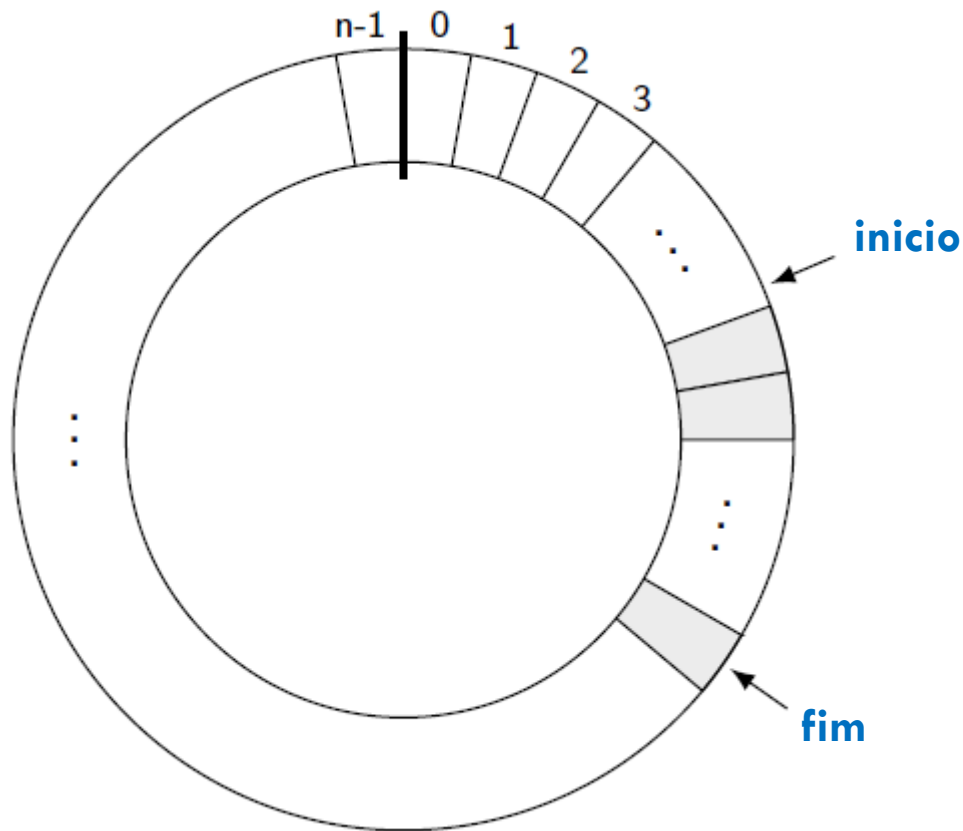
28

- Evita o deslocamento dos elementos
 - ▣ Necessita dos índices *inicio* e *fim* na definição de Fila
- Representação ainda ineficiente
 - ▣ Situação em que a fila está vazia, mas nenhum elemento novo pode ser inserido
- Proposta de melhoria
 - ▣ Visualizar o vetor que armazena a fila como um círculo

Fila sequencial circular

29

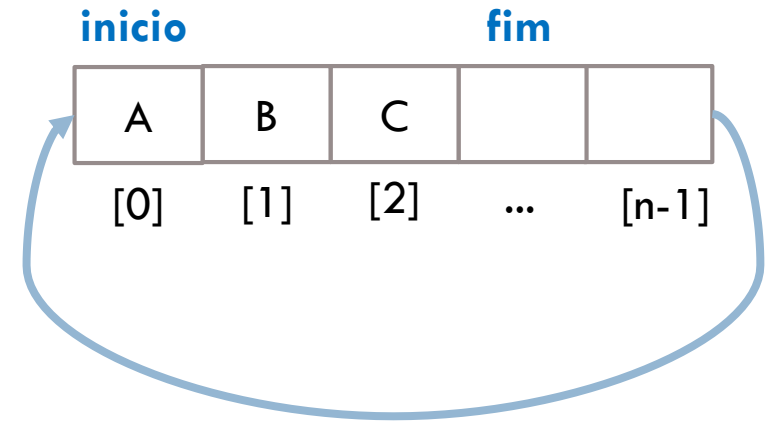
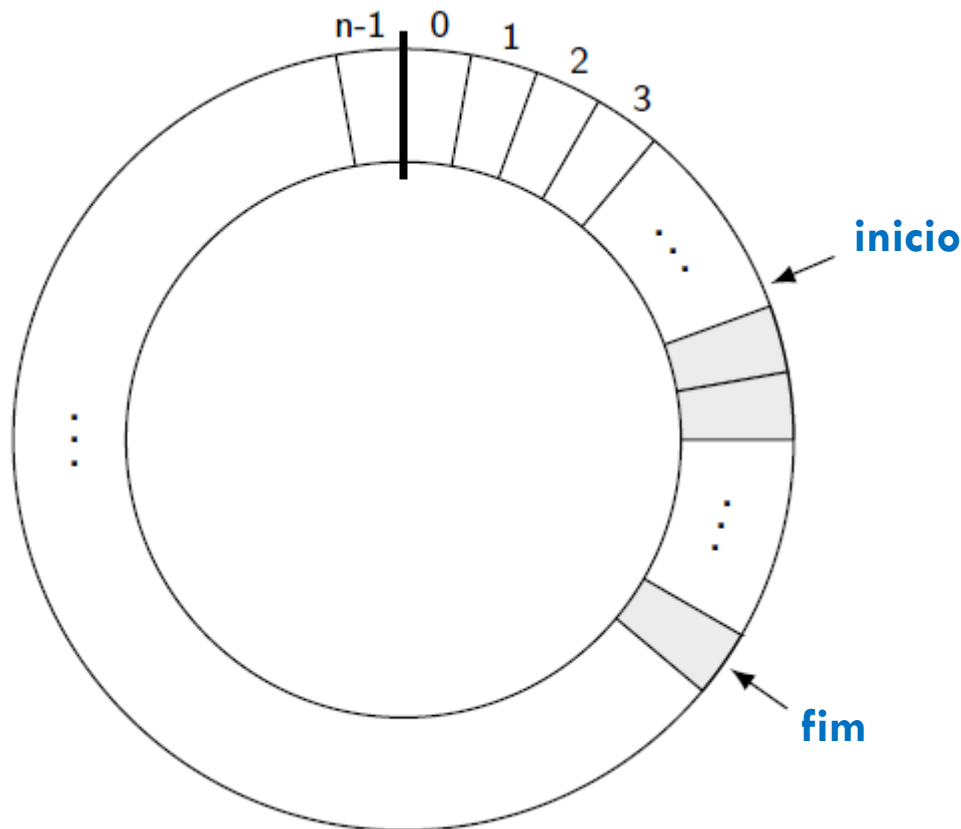
- Vetor de tamanho N considerado de maneira circular



Fila sequencial circular

30

- Vetor de tamanho N considerado de maneira circular



- As manipulações de índices **inicio** e **fim** são realizadas segundo o tamanho máximo (N) do vetor

Estrutura da fila sequencial circular

31

```
typedef struct Fila {  
    int *elems; //alocação de vetor para armazenar os dados  
    //int elems[T_MAX] [#define T_MAX 50;]  
    int inicio;  
    int fim;  
    int qtd; //quantidade de elementos presente na fila  
    int QTD_MAX; //tamanho máximo do vetor  
} Fila;
```

Modificações no enfileiramento

32

- Em relação ao índice *fim* (final da fila)
 - ▣ Próxima posição do vetor não será $fim = fim + 1$
 - ▣ Final da fila é dado por:
 - $fim = (fim + 1) \% QTD\ MAX$
- ▣ Uso do **resto da divisão** garante que ao chegar ao final do vetor a próxima posição de inserção será o índice 0 (zero) do vetor

Modificações no desenfileiramento

33

- Mesmo princípio é usado para gerenciar o índice de *inicio* da fila
- Índice *inicio*
 - ▣ Nova posição do inicio da fila não é dado por:
 - $\text{inicio} = \text{inicio} + 1$
 - ▣ Inicio da fila é dado por:
 - $\text{inicio} = (\text{inicio} + 1) \% \text{QTD MAX}$

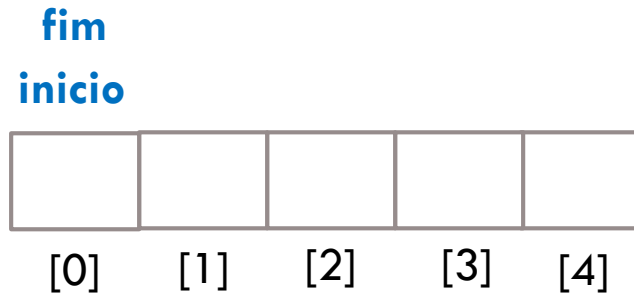
Exemplo de fila sequencial circular

34

□ Fila de tamanho máximo igual a 5

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=0
Fim=0
Qtd=0

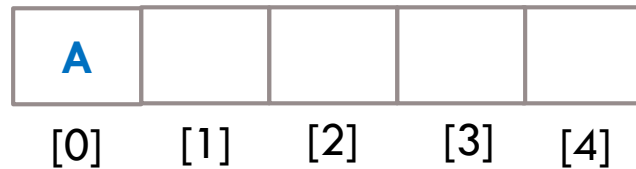
Exemplo de fila sequencial circular

35

QTD_MAX=5

- **enfileirar(A)**
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)

inicio fim



Inicio=0

Fim=1

Qtd=1

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

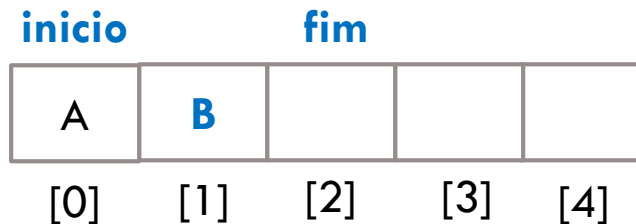
$\text{fim} = (0 + 1) \% 5 \Rightarrow 1$

Exemplo de fila sequencial circular

36

QTD_MAX=5

- enfileirar(A)
- **enfileirar(B)**
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=0

Fim=2

Qtd=2

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

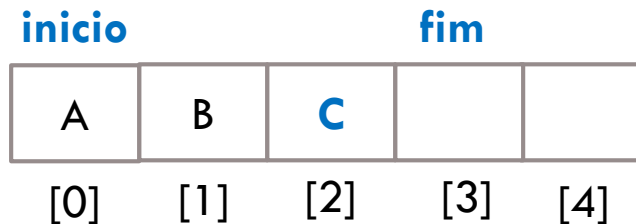
$\text{fim} = (1 + 1) \% 5 \Rightarrow 2$

Exemplo de fila sequencial circular

37

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- **enfileirar(C)**
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=0

Fim=3

Qtd=3

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

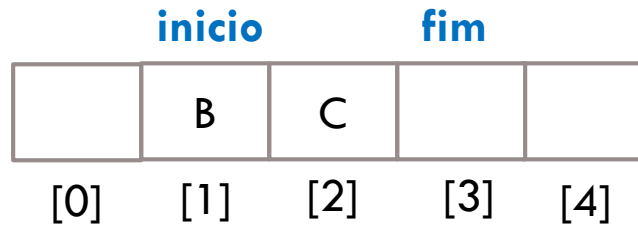
$\text{fim} = (2 + 1) \% 5 \Rightarrow 3$

Exemplo de fila sequencial circular

38

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- **desenfileirar()**
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=1
Fim=3
Qtd=2

$\text{inicio} = (\text{inicio} + 1) \% \text{QTD_MAX}$

$\text{inicio} = (0 + 1) \% 5 \Rightarrow 1$

Exemplo de fila sequencial circular

39

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- **enfileirar(D)**
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=1

Fim=4

Qtd=3

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

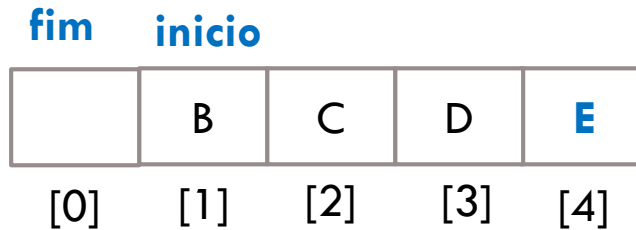
$\text{fim} = (3 + 1) \% 5 \Rightarrow 4$

Exemplo de fila sequencial circular

40

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- **enfileirar(E)**
- desenfileirar()
- enfileirar(F)
- enfileirar(G)



Inicio=1

Fim=0

Qtd=4

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

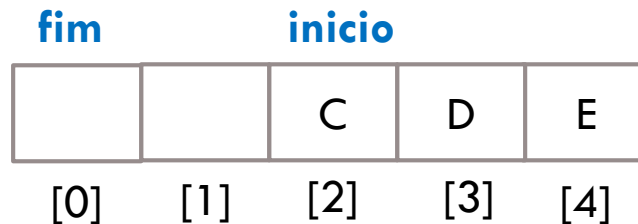
$\text{fim} = (4 + 1) \% 5 \Rightarrow 0$

Exemplo de fila sequencial circular

41

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- **desenfileirar()**
- enfileirar(F)
- enfileirar(G)



Inicio=2

Fim=0

Qtd=3

$\text{inicio} = (\text{inicio} + 1) \% \text{QTD_MAX}$

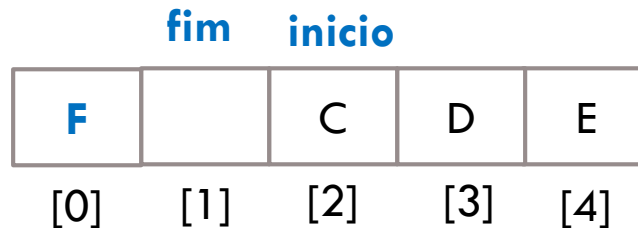
$\text{inicio} = (1 + 1) \% 5 \Rightarrow 2$

Exemplo de fila sequencial circular

42

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- **enfileirar(F)**
- enfileirar(G)



Inicio=2

Fim=1

Qtd=4

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

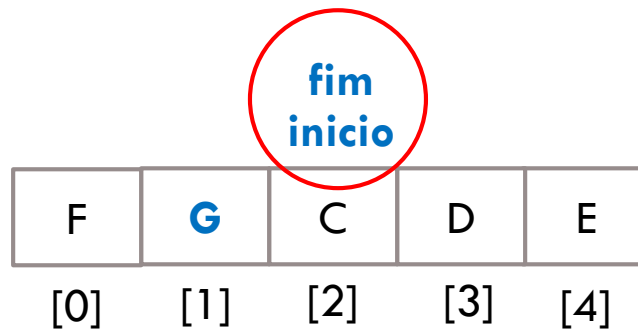
$\text{fim} = (0 + 1) \% 5 \Rightarrow 1$

Exemplo de fila sequencial circular

43

QTD_MAX=5

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()
- enfileirar(D)
- enfileirar(E)
- desenfileirar()
- enfileirar(F)
- **enfileirar(G)**



Inicio=2

Fim=2

Qtd=5

$\text{fim} = (\text{fim} + 1) \% \text{QTD_MAX}$

$\text{fim} = (1 + 1) \% 5 \Rightarrow 2$

Operações na fila sequencial circular

45

- ❑ Inicialização
- ❑ Destruição
- ❑ Fila cheia
- ❑ Fila vazia
- ❑ Inserção (enfileirar)
- ❑ Remoção (desenfileirar)
- ❑ Consultar elemento da fila

Operações na fila sequencial circular

46

□ Inicialização

```
Fila* inicializarFila(int tam){  
    Fila *f = malloc(sizeof(Fila);  
    f->elems=malloc(sizeof(int*tam));  
    f->inicio=0; f->fim=0; f->qtd=0;  
    f->QTD_MAX=tam;  
    return f; }
```

Operações na fila sequencial circular

47

□ Inicialização

```
Fila* inicializarFila(int tam){  
    Fila *f = malloc(sizeof(Fila));  
    f->elems=malloc(sizeof(int*tam));  
    f->inicio=0; f->fim=0; f->qtd=0;  
    f->QTD_MAX=tam;  
    return f; }
```

□ Destruição

```
void finalizarFila(Fila *f){  
    free(f->elems);  
    free(f);  
}
```

Operações na fila sequencial circular

48

□ Fila Cheia

```
bool filaCheia(Fila *f){  
    if (f->qtd==f->QTD_MAX) return true; else return false;  
}
```

Operações na fila sequencial circular

49

□ Fila Cheia

```
bool filaCheia(Fila *f){  
    if (f->qtd==f->QTD_MAX) return true; else return false;  
}
```

□ Fila Vazia

```
bool filaVazia(Fila *f){  
    if (f->qtd==0) return true; else return false;  
}
```


Operações na fila sequencial circular

50

□ Inserção (enfileiramento)

```
bool enfileirar(Fila *f, int valor){  
    if (!filaCheia(f)) {  
        f->elems[f->fim] = valor;  
        f->fim = (f->fim + 1) % f->QTD_MAX;  
        (f->qtd)++;  
        return true;  
    } return false;  
}
```

Operações na fila sequencial circular

51

□ Inserção (enfileiramento)

```
bool enfileirar(Fila *f, int valor){  
    if (!filaCheia(f)) {  
        f->elems[f->fim] = valor;  
        f->fim = (f->fim + 1) % f->QTD_MAX;  
        (f->qtd)++;  
        return true;  
    } return false;  
}
```

É possível implementar esse procedimento sem o índice *fim*?
Como?

Operações na fila sequencial circular

52

□ Inserção (enfileiramento) - **sem o índice fim**

```
bool enfileirar(Fila *f, int valor){
    if (!filaCheia(f)) {
        int posicao = (f->inicio + f->qtd) % f->QTD_MAX;
        f->elems[posicao] = valor;
        (f->qtd)++;
        return true;
    } return false;
}
```

Operações na fila sequencial circular

53

□ Remoção (desenfileiramento)

```
bool desenfileirar(Fila *f, int *elem){
    if (!filaVazia(f){
        (*elem) = f->elems[f->inicio]; //retorna elemento
        f->inicio = (f-> inicio + 1) % f->QTD_MAX;
        (f->qtd)--;
        return true;
    }
    return false;
}
```

Operações na fila sequencial circular

54

□ Consultar elemento na fila

```
int consultarFila(Fila *f){  
    if(!FilaVazia(f)) {  
        return f->elems[f->inicio];  
    }  
}
```

Usando as operações da fila circular

55

```
int main(){  
    int tam, pos, valor;  
    printf("Digite o tamanho da fila: "); scanf("%d", &tam);  
    Fila f* = inicializarFila(tam);
```

Usando as operações da fila circular

56

```
int main(){
    int tam, pos, valor;
    printf("Digite o tamanho da fila: "); scanf("%d", &tam);
    Fila f* = inicializarFila(tam);
    for(pos=0; pos<tam;pos++)
        enfileirar(f, pos);
```

Usando as operações da fila circular

57

```
int main(){
    int tam, pos, valor;
    printf("Digite o tamanho da fila: "); scanf("%d", &tam);
    Fila f* = inicializarFila(tam);
    for(pos=0; pos<tam;pos++)
        enfileirar(f, pos);
    while(!filaVazia(f)){
        desifileirar(f, &valor);
        printf("valor = %d\n", valor);
    }
    finalizarFila(f);
    return (0);
}
```


Usando as operações da fila circular

58

```
int main(){
    int tam, pos, valor;
    printf("Digite o tamanho da fila: "); scanf("%d", &tam);
    Fila f* = inicializarFila(tam);
    for(pos=0; pos<tam*2;pos++)
        enfileirar(f, pos);
    while(!filaVazia(f)){
        desifileirar(f, &valor);
        printf("valor = %d\n", valor);
    }
    finalizarFila(f);
    return (0);
}
```

Qual o problema
com esse código?
Como resolver?

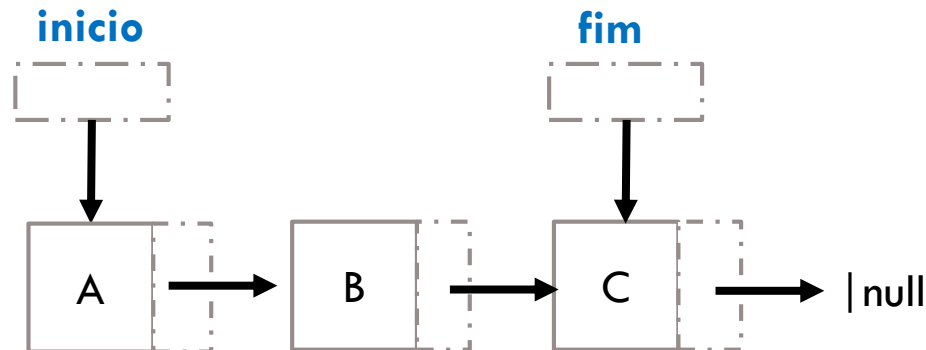
59

Implementação de fila com estruturas ligadas

Implementação com lista ligada

60

- ❑ Uso de ponteiros entre nós
- ❑ Não necessita de uma porção de memória contígua alocada a priori
- ❑ Alocação de elementos feita sob demanda

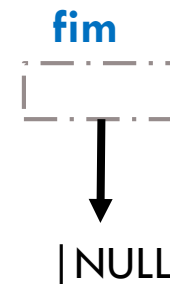
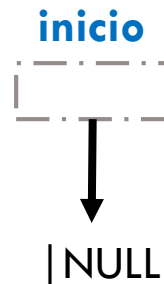


Exemplo de Fila (como lista ligada)

61

□ Fila vazia **inicio** e **fim** aponta para *null*

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- desenfileirar()

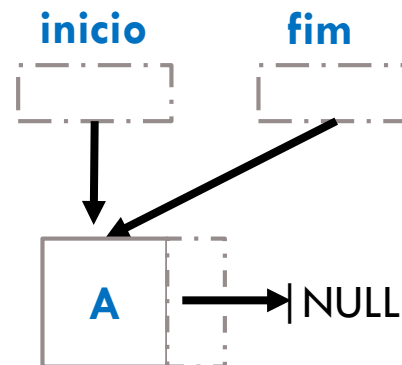


Exemplo de Fila (como lista ligada)

62

□ Enfileiramento inseri elemento no final da lista

- **enfileirar(A)**
- enfileirar(B)
- enfileirar(C)
- desenfileirar()

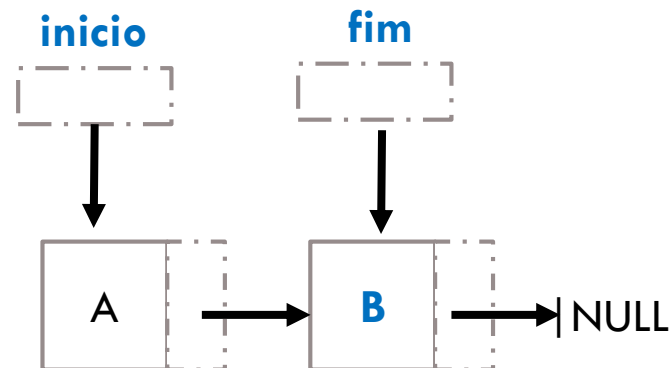


Exemplo de Fila (como lista ligada)

63

□ Enfileiramento inseri elemento no final da lista

- enfileirar(A)
- **enfileirar(B)**
- enfileirar(C)
- desenfileirar()

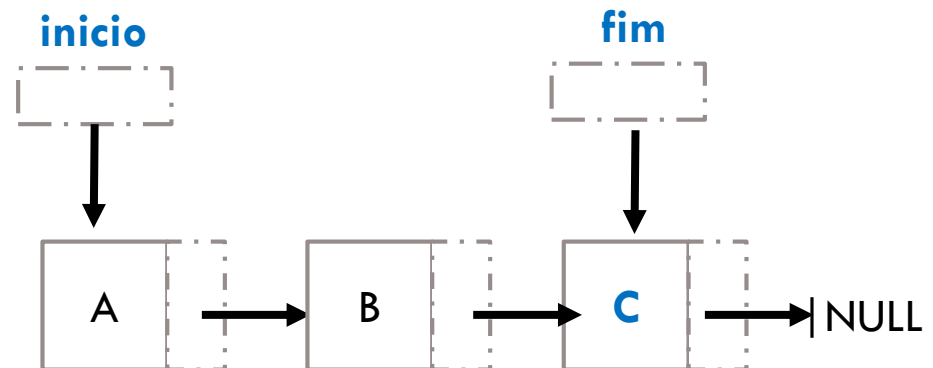


Exemplo de Fila (como lista ligada)

64

□ Enfileiramento inseri elemento no final da lista

- enfileirar(A)
- enfileirar(B)
- **enfileirar(C)**
- desenfileirar()

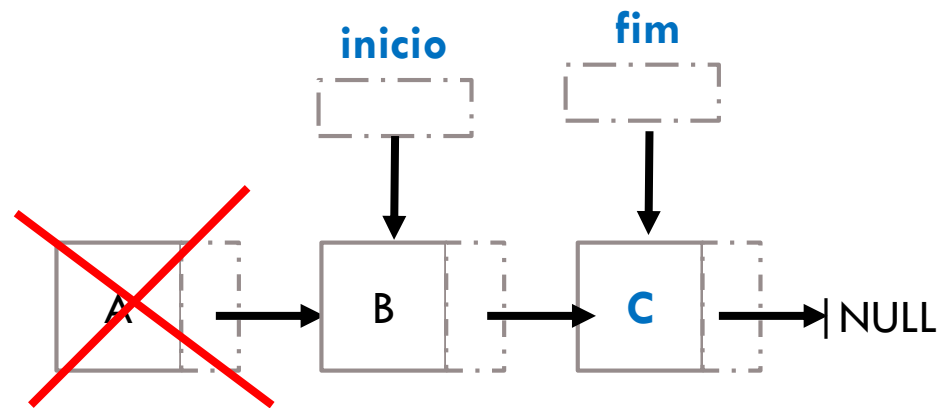


Exemplo de Fila (como lista ligada)

65

□ Desenfileiramento remove elemento no **inicio** da lista

- enfileirar(A)
- enfileirar(B)
- enfileirar(C)
- **desenfileirar()**



Estrutura da Fila (como lista ligada)

66

Elemento nó para a fila

```
typedef struct NoFila{  
    int chave;  
    struct NoFila *prox;  
} NoFila;
```

Fila

```
typedef struct Fila {  
    NoFila *inicio;  
    NoFila *fim;  
} Fila;
```

Operações da Fila (como lista ligada)

67

□ Inicialização

```
void inicializarFila(Fila *f){  
    f->inicio=NULL;  
    f->fim=NULL;  
}
```

Operações da Fila (como lista ligada)

68

□ Inserção (no fim da lista)

```
bool enfileirar(Fila *f, int valor){
    NoFila* novo=malloc(sizeof(NoFila));
    if (novo!=NULL){
        novo->chave=valor;
        novo->prox=NULL;
        if (f->fim!=NULL) //não vazia
            f->fim->prox=novo;
        else
            f->inicio=novo; //primeiro nó inserido
        f->fim=novo; return true;
    } return false;
}
```

Operações da Fila (como lista ligada)

69

□ Remoção (no início da lista)

```
bool desenfileirar(Fila *f, int *valor){
    NoFila *aux = f->inicio;
    if (aux==NULL) //fila vazia
        return false;
    *valor= aux->chave;
    f->inicio=f->inicio->prox; //caminha na lista
    if (f->inicio == NULL) //fila ficou vazia
        f->fim = NULL;
    free(aux);
    return true;
}
```

Operações da Fila (como lista ligada)

70

□ Consulta Fila

```
int consultarFila(Fila *f){  
    if (f!=NULL)  
        return f->inicio->chave;  
}
```

Operações da Fila (como lista ligada)

71

□ Consulta Fila

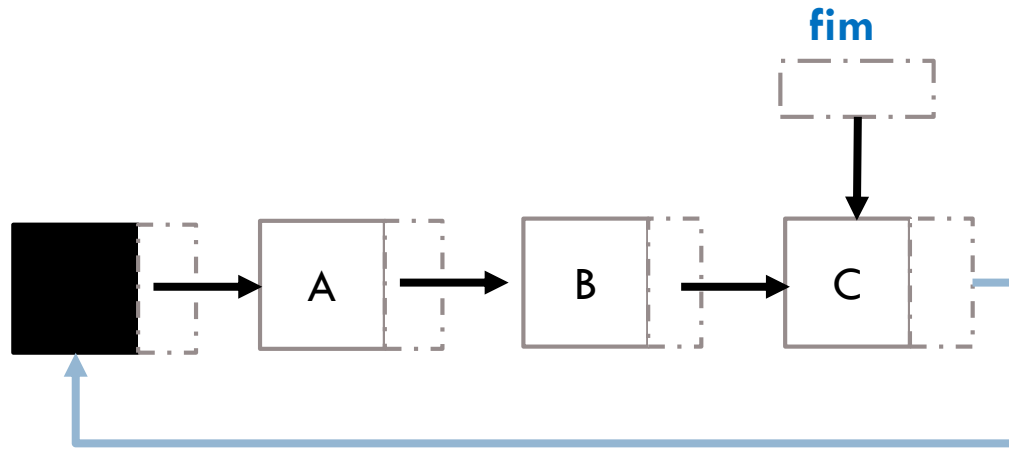
```
int consultarFila(Fila *f){  
    if (f!=NULL)  
        return f->inicio->chave;  
}
```

□ Fila Vazia

```
bool filaVazia(Fila *f){  
    if (f!=NULL && f->inicio==NULL)  
        return true;  
    else  
        return false;  
}
```

Fila como uma lista ligada circular

72



- Facilita a representação de filas
 - ▣ Uso de apenas um ponteiro para o fim da lista

Exercício

73

- Implemente as principais operações de uma fila usando uma lista ligada circular

Exemplo de aplicações de fila

76

- Fila de impressão
- Escalonamento de processos em sistemas operacionais
- Percurso em estrutura de dados não lineares (e.g., grafos)
- Comunicação entre computadores (redes)

Síntese

77

- Filas são estruturas lineares no qual o primeiro elemento a entrar é o primeiro a sair
 - ▣ Elemento removido é aquele que está há mais tempo

Síntese

78

- Filas são estruturas lineares no qual o primeiro elemento a entrar é o primeiro a sair
 - ▣ Elemento removido é aquele que está há mais tempo
- Estudamos a definição da estrutura e as operações da implementação com arranjos e com listas ligadas
- Verificamos a opção de fila circular tanto como vetores como em lista ligada