

Aula 17 - Filas

Implementação: Estática/Sequencial

Prof. Me. Claudiney R. Tinoco

`profclaudineytinoco@gmail.com`

Faculdade de Computação (FACOM)
Bacharelado em Ciência da Computação (BCC)
Bacharelado em Sistemas de Informação (BSI)

Algoritmos e Estruturas de Dados 1 (AED1)
GBC024 - GSI006



Introdução

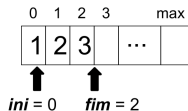
- Aloca posição para todos os elementos quando a estrutura é criada
 - Utiliza um **vetor com *max* posições**

0	1	2	3		max
1	2	3		...	



Introdução

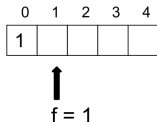
- Aloca posição para todos os elementos quando a estrutura é criada
 - Utiliza um **vetor com *max* posições**
- Eficiência depende do acesso rápido às extremidades da fila
 - Inserção precisa conhecer o **final da fila**
 - Remoção precisa conhecer o **início da fila**





Formas de Implementação

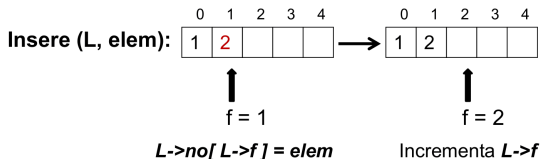
- **Ideia 1: utilizar uma lista linear simples**
 - **Início fixo** na posição ZERO (dispensa campo)





Formas de Implementação

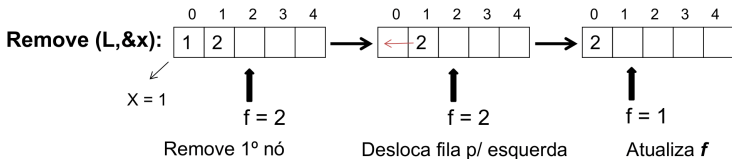
- **Ideia 1: utilizar uma lista linear simples**
 - **Início fixo** na posição ZERO (dispensa campo)
 - Inserir no final é uma **operação simples**





Formas de Implementação

- **Ideia 1: utilizar uma lista linear simples**
 - **Início fixo** na posição ZERO (dispensa campo)
 - Inserir no final é uma **operação simples**
 - **Problema:** remoção envolve **deslocamento de todos os elementos restantes (ineficiente)**





Formas de Implementação

- **Ideia 2: utilizar uma lista linear circular**
 - Usa campo *ini* para indicar 1º nó da fila
 - Adota um **incremento circular** ($x = x \oplus 1$):

$$x = \begin{cases} x + 1, & \text{se } x+1 < \text{max} \\ 0, & \text{se } x+1 = \text{max} \end{cases}$$

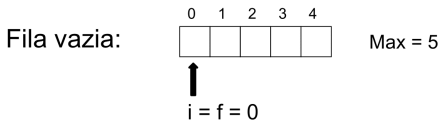
OU

$$x = (x+1) \% \text{max}$$



Fila Circular

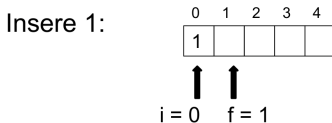
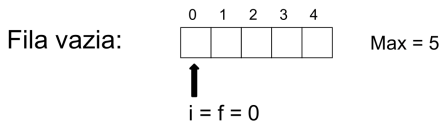
Exemplo:





Fila Circular

Exemplo:



$$\text{Vet}[f] = E$$
$$f = f \oplus 1$$

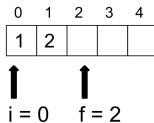
Incremento Circular



Fila Circular

Exemplo:

Insere 2:



$$\text{Vet}[f] = E$$
$$f = f \oplus 1$$



Fila Circular

Exemplo:

Insere 2:

0	1	2	3	4
1	2			



$i = 0$



$f = 2$

$Vet[f] = E$
 $f = f \oplus 1$

Insere 3:

0	1	2	3	4
1	2	3		



$i = 0$



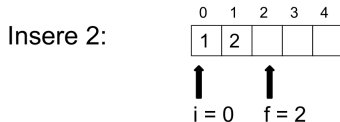
$f = 3$

$Vet[f] = E$
 $f = f \oplus 1$



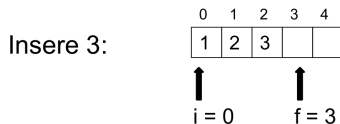
Fila Circular

Exemplo:



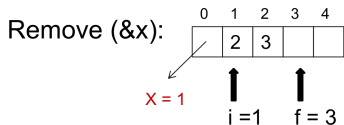
$$\text{Vet}[f] = E$$

$$f = f \oplus 1$$



$$\text{Vet}[f] = E$$

$$f = f \oplus 1$$



$$E = \text{Vet}[i];$$

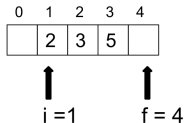
$$i = i \oplus 1$$



Fila Circular

Exemplo:

Insere 5:

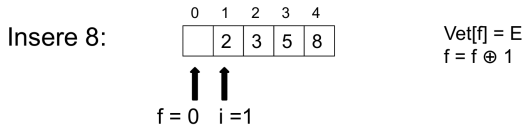
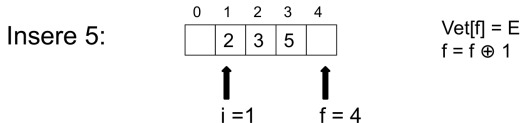


Vet[f] = E
 $f = f \oplus 1$



Fila Circular

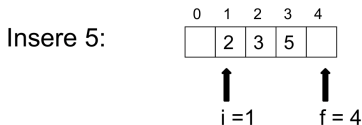
Exemplo:



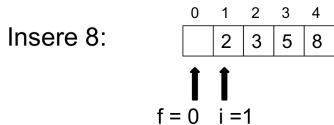


Fila Circular

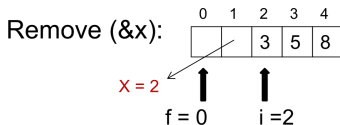
Exemplo:



$Vet[f] = E$
 $f = f \oplus 1$



$Vet[f] = E$
 $f = f \oplus 1$



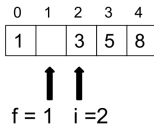
$E = Vet[i];$
 $i = i \oplus 1$



Fila Circular

Exemplo:

Insere 1:

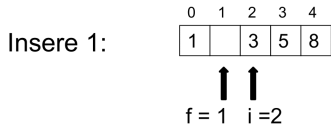


Vet[f] = E
 $f = f \oplus 1$

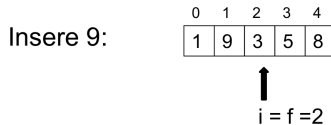


Fila Circular

Exemplo:



$$\text{Vet}[f] = E$$
$$f = f \oplus 1$$

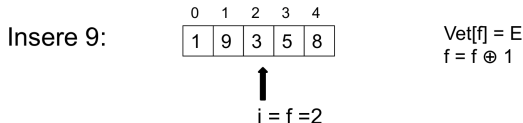
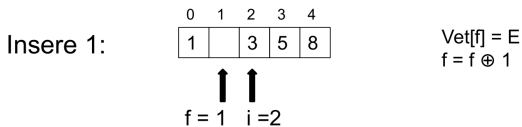


$$\text{Vet}[f] = E$$
$$f = f \oplus 1$$



Fila Circular

Exemplo:



- **Problema:** Diferenciar fila vazia e fila cheia
 - Ambos casos são $ini = fim$



Fila Circular

- Existem **2 soluções**:
 - Abordagem 1: desperdício de 1 posição
 - Abordagem 2: uso de um contador



Fila com Desperdício de Posição

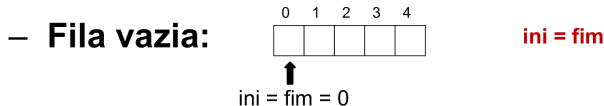
- São armazenados ***max-1*** elementos
 - Último nó da fila fica sempre vazio





Fila com Desperdício de Posição

- São armazenados ***max-1*** elementos
 - Último nó da fila fica sempre vazio

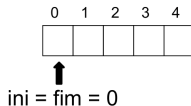




Fila com Desperdício de Posição

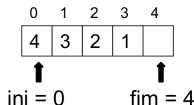
- São armazenados ***max-1*** elementos
 - Último nó da fila fica sempre vazio

– Fila vazia:



ini = fim

– Fila cheia:

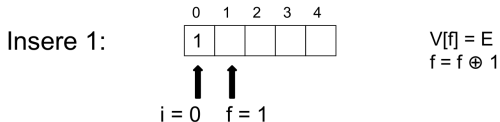
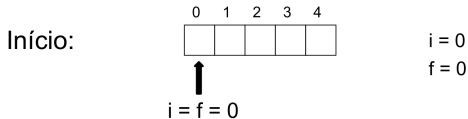


ini = fim \oplus 1



Fila com Desperdício de Posição

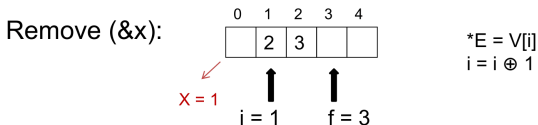
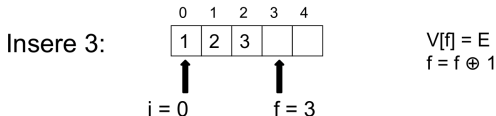
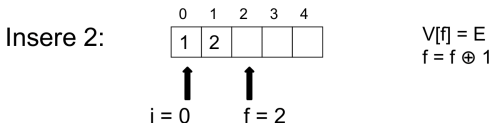
- Exemplo:





Fila com Desperdício de Posição

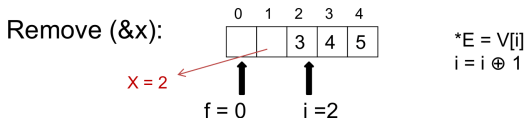
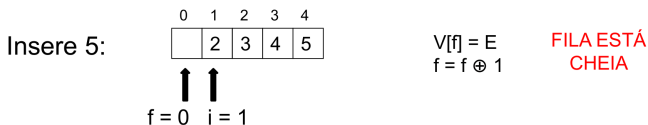
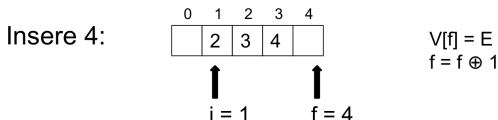
Exemplo:





Fila com Desperdício de Posição

Exemplo:

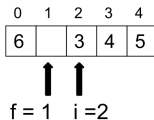




Fila com Desperdício de Posição

- Exemplo:

Inserir 6:



$$V[f] = E$$
$$f = f \oplus 1$$

FILA ESTÁ
CHEIA

Inserir 7:

Operação falha
(FILA CHEIA)



Fila com Desperdício de Posição

- **Estrutura de representação:**
 - Vetor com *max* posições



Fila com Desperdício de Posição

- **Estrutura de representação:**
 - Vetor com *max* posições
 - Campo para indicar o início da fila (*ini*)



Fila com Desperdício de Posição

- **Estrutura de representação:**
 - Vetor com *max* posições
 - Campo para indicar o início da fila (*ini*)
 - Campo para indicar o fim da fila (*fim*)



Fila com Desperdício de Posição

- **Estrutura de representação:**
 - Vetor com **max** posições
 - Campo para indicar o início da fila (**ini**)
 - Campo para indicar o fim da fila (**fim**)

Exemplo: **fila de inteiros**

fila.c

```
# define max 20
struct fila {
    int vetor [max];
    int ini, fim;
};
```

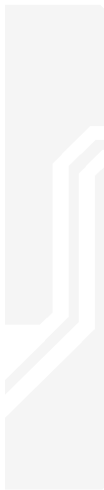
fila.h

```
typedef struct fila * Fila;
```



Fila com Desperdício de Posição

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)





Fila com Desperdício de Posição

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)
 - *Coloca a fila no estado de vazia (**ini = fim**)*
 - Inicializados como **ZERO** ou **max-1**



Fila com Desperdício de Posição

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)
 - Coloca a fila no estado de vazia (*ini = fim*)
 - Inicializados como **ZERO** ou *max-1*

```
Fila cria_fila() {  
    Fila f;  
    f = (Fila) malloc(sizeof(struct fila));  
    if (f != NULL) {  
        f->ini = 0;  
        f->fim = 0;  
    }  
    return f;  
}
```



Fila com Desperdício de Posição

- Operação **fila_vazia()**:
 - Verifica se a fila está no estado de vazia (*ini=fim*)
 - *Independente do valor atual dos campos (não precisa ter o mesmo valor usado na inicialização)*

```
int fila_vazia(Fila f) {  
    if (f->ini == f->fim)  
        return 1;  
    else  
        return 0;  
}
```



Fila com Desperdício de Posição

- Operação **fila_cheia()**:
 - Verifica se a fila está no estado de cheia
(**$ini = fim \oplus 1$**)

```
int fila_cheia(Fila f) {  
    if (f->ini == (f->fim+1)%max)  
        return 1;  
    else  
        return 0;  
}
```



Fila com Desperdício de Posição

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - Posição indicada pelo campo *fim*





Fila com Desperdício de Posição

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - Posição indicada pelo campo ***fim***
 - Incrementa o campo ***fim*** (**incremento circular**)



Fila com Desperdício de Posição

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - Posição indicada pelo campo **fim**
 - Incrementa o campo **fim** (**incremento circular**)

```
int insere_fim(Fila f, int elem) {  
    if (fila_cheia(f) == 1)  
        return 0;  
    // Insere elemento no final  
    f->no[f->fim] = elem;  
    f->fim = (f->fim+1)%max; // Incremento circular  
    return 1;  
}
```



Fila com Desperdício de Posição

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*





Fila com Desperdício de Posição

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*
 - Incrementa o campo *ini* (**incremento circular**)



Fila com Desperdício de Posição

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo **ini**
 - Incrementa o campo **ini** (**incremento circular**)

```
int remove_ini(Fila f, int *elem) {  
    if (fila_vazia(f) == 1)  
        return 0;  
    // Remove o elemento do inicio  
    *elem = f->no[f->ini];  
    f->ini = (f->ini+1)%max; // Incremento circular  
    return 1;  
}
```



Fila com Uso do Contador

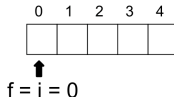
- Utiliza um campo para contar elementos
 - **Fim** obtido a partir do **início** e do **contador**
(**$fim = ini \oplus cont = (ini + cont) \% max$**)



Fila com Uso do Contador

- Utiliza um campo para contar elementos
 - **Fim** obtido a partir do **início** e do **contador**
(**fim** = **ini** \oplus **cont** = (**ini** + **cont**) % **max**)

– Fila vazia:



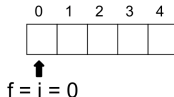
cont = 0



Fila com Uso do Contador

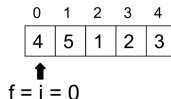
- Utiliza um campo para contar elementos
 - Fim** obtido a partir do **início** e do **contador**
($fim = ini \oplus cont = (ini + cont) \% max$)

– Fila vazia:



cont = 0

– Fila cheia:

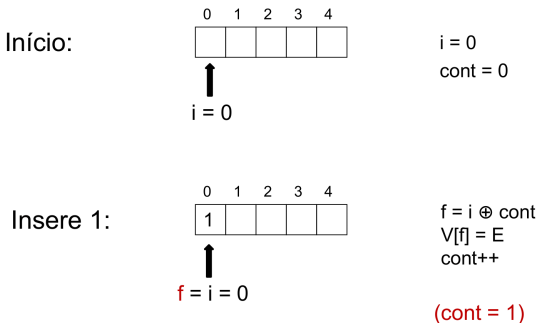


cont = max



Fila com Uso do Contador

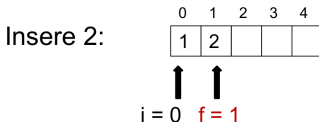
- Exemplo:





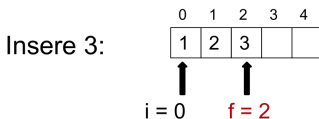
Fila com Uso do Contador

Exemplo:



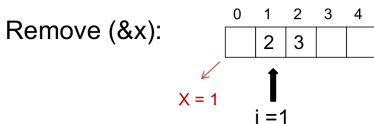
$f = i \oplus \text{cont}$
 $V[f] = E$
 $\text{cont}++$

(cont = 2)



$f = i \oplus \text{cont}$
 $V[f] = E$
 $\text{cont}++$

(cont = 3)



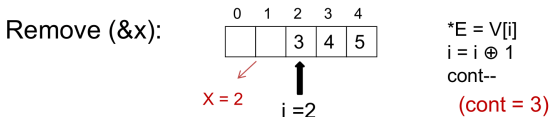
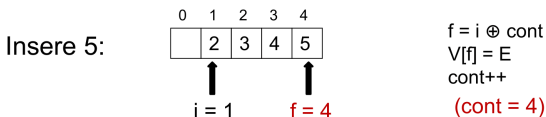
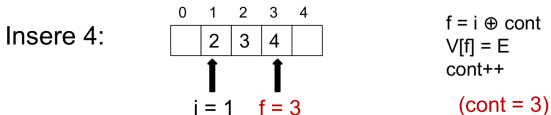
$*E = V[i]$
 $i = i \oplus 1$
 $\text{cont}--$

(cont = 2)



Fila com Uso do Contador

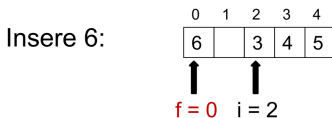
Exemplo:





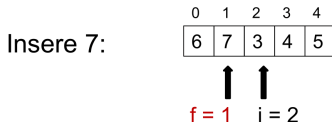
Fila com Uso do Contador

- Exemplo:



$f = i \oplus \text{cont}$
 $V[f] = E$
 $\text{cont}++$

(cont = 4)



$f = i \oplus \text{cont}$
 $V[f] = E$
 $\text{cont}++$

(cont = 5)

Inserir 8: Operação falha (**FILA ESTÁ CHEIA**)



Fila com Uso do Contador

- **Estrutura de representação:**
 - Vetor com *max* posições



Fila com Uso do Contador

- **Estrutura de representação:**
 - Vetor com *max* posições
 - Campo para indicar o início da fila (*ini*)



Fila com Uso do Contador

- **Estrutura de representação:**
 - Vetor com *max* posições
 - Campo para indicar o início da fila (*ini*)
 - Campo para contar os elementos (*cont*)
 - *Fim* obtido a partir dos campos *ini* e *cont*



Fila com Uso do Contador

- **Estrutura de representação:**
 - Vetor com *max* posições
 - Campo para indicar o início da fila (*ini*)
 - Campo para contar os elementos (*cont*)
 - *Fim* obtido a partir dos campos *ini* e *cont*

Exemplo: fila de inteiros

fila.c

```
# define max 20
struct fila {
    int vetor [max];
    int ini, cont;
};
```

fila.h

```
typedef struct fila * Fila;
```



Fila com Uso do Contador

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)





Fila com Uso do Contador

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)
 - *Coloca a fila no estado de vazia*
 - **Ini** e **cont** inicializados com **ZERO**



Fila com Uso do Contador

- Operação **cria_fila()**:
 - Aloca todas as posições da estrutura fila (**vetor**)
 - *Coloca a fila no estado de vazia*
 - **ini** e **cont** inicializados com **ZERO**

```
Fila cria_fila() {  
    Fila f;  
    f = (Fila) malloc(sizeof(struct fila));  
    if (f != NULL) {  
        f->ini = 0;  
        f->cont = 0;  
    }  
    return f;  
}
```



Fila com Uso do Contador

- Operação **fila_vazia()**:
 - Verifica se a fila está no estado de vazia (**cont=0**)
 - *Independente dos valores dos campos **ini** e **fim***

```
int fila_vazia(Fila f) {  
    if (f->cont == 0)  
        return 1;  
    else  
        return 0;  
}
```




Fila com Uso do Contador

- Operação **fila_cheia()**:
 - Verifica se a fila está cheia (**cont = max**)
 - **max** corresponde a quantidade máxima de elementos permitidos na fila

```
int fila_cheia(Fila f) {  
    if (f->cont == max)  
        return 1;  
    else  
        return 0;  
}
```



Fila com Uso do Contador

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - **Posição calculada** a partir dos campos *ini* e *cont*



Fila com Uso do Contador

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - **Posição calculada** a partir dos campos *ini* e *cont*
 - Incrementa (**NÃO circular**) o campo *cont*



Fila com Uso do Contador

- Operação **insere_fim()**:
 - Insere o elemento no final da fila
 - **Posição calculada** a partir dos campos **ini** e **cont**
 - Incrementa (**NÃO circular**) o campo **cont**

```
int insere_fim(Fila f, int elem) {  
    if (fila_cheia(f) == 1)  
        return 0;  
    // Insere elemento no final  
    f->no[(f->ini+f->cont)%max] = elem;  
    f->cont++; // Incremento normal  
    return 1;  
}
```



Fila com Uso do Contador

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*





Fila com Uso do Contador

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*
 - Incrementa o campo *ini* (**incremento circular**)



Fila com Uso do Contador

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*
 - Incrementa o campo *ini* (**incremento circular**)
 - Decrementa o campo *cont*



Fila com Uso do Contador

- Operação **remove_ini()**:
 - Retorna o valor do elemento no início da fila
 - Posição indicada pelo campo *ini*
 - Incrementa o campo *ini* (**incremento circular**)
 - Decrementa o campo *cont*

```
int remove_ini(Fila f, int *elem) {  
    if (fila_vazia(f) == 1) return 0;  
    // Remove o elemento do inicio  
    *elem = f->no[f->ini];  
    f->ini = (f->ini+1)%max; // Incremento circular  
    f->cont--; // Decremento não circular  
    return 1; }
```




Referências

✓ Básica

- CELES, W., CERQUEIRA, R. e RANGEL, J. L. *“Introdução a estruturas de dados”*. Campus Elsevier, 2004.
- TENENBAUM, A. M., LANGSAM, Y. e AUGENSTEIN, M.J. *“Estrutura de Dados Usando C”*. Makron Books.

✓ Extra

- BACKES, André. *“Programação Descomplicada Linguagem C”*. Projeto de extensão que disponibiliza vídeo-aulas de C e Estruturas de Dados. Disponível em: <https://www.youtube.com/user/progdescomplicada>. Acessado em: 25/04/2022.

✓ Baseado nos materiais dos seguintes professores:

- Prof. André Backes (UFU)
- Prof. Bruno Travençolo (UFU)
- Prof. Luiz Gustavo de Almeida Martins (UFU)

Dúvidas?

Prof. Me. Claudiney R. Tinoco
profclaudineytinoco@gmail.com

Faculdade de Computação (FACOM)
Universidade Federal de Uberlândia (UFU)