

# Módulo 5

## *Filas*

Métodos e Algoritmos  
Computacionais

C++

(Rone Ilídio)

# Filas

- Sequência de elementos na memória
- Implementação
  - Vetor
  - Lista
- Funções
  - Inserção: sempre no final
  - Remoção: o primeiro a chegar, primeiro a sair
  - Imprime
- Política FIFO: First In First Out

# Filas - Vetor

- Vetor de 0 a  $n-1$
- Fila:
  - Posição inicial  $s$      $0 \leq s < n$
  - Posição final  $t$      $0 \leq t < n$
  - Obs:  $t$  sempre aponta para uma posição vazia
- Fila cheia
  - $t+1 = s$     ou     $t+1 = n$  e  $s=0$
  - $(t+1)\%n = s$
- Fila vazia
  - $t = s$

# Fila – Estrutura de Dados

```
struct cel{  
    int conteudo;  
};  
typedef cel celula;  
int const n = 4;  
celula f[n];  
int s=0, t=0;
```

# Fila – Implementação em Vetor

- Função main (exemplo)

```
int main(){  
    insere(5);insere(3);  
    insere(2); insere(4);  
    imprime();  
    celula r = remove();  
    cout << "\nRemovido:" <<r.conteudo;  
    imprime();  
    getch();  
}
```

# Fila – Implementação em Vetor

- Insere → se na fila não estiver cheia
  - $f[t] = y$ ;
  - $t++$

```
void insere(int y){  
    if((t+1)%n == s)                //Pilha cheia  
        cout << "\nPilha cheia";  
    else {  
        f[t].conteudo = y;  
        t++;  
        if (t == n) t = 0;  
    }  
}
```

# Fila – Implementação em Vetor

- Remove → se a fila não estiver vazia ( $t \neq s$ )

```
celula remove(){  
    celula x;  
    x.conteudo = 0; //valor nulo  
    if (t!=s){  
        x = f[s];  
        s++;  
        if (s == n) s=0;  
    }  
    else {  
        cout << "\nPilha Vazia!";  
    }  
    return x;  
}
```

```
void imprime(){
    cout << "\nFila:(s="<<s<<", t="<<t<<"): ";
    int a = s;
    while(a!=t){
        cout<<" "<<f[a].conteudo;
        a++;
        if(a==n) a=0;
    }
}
```

Fila – Com Lista Encadeada

# Fila – Com Lista Encadeada

- Tamanho ilimitado da fila
- Mesmas operações
  - Insere
  - Remove

# Fila – Com Lista Encadeada

- Inserir
- Não precisa verificar se a pilha está cheia
- Função insere: (próximo slide)

# Exemplo de função main

```
int main(){
    celula *s, *t;
    s = t = NULL; //Fila Vazia
    insere(5,&s,&t);insere(3,&s,&t);
    insere(2,&s,&t);insere(4,&s,&t);
    imprime(s);
    int r = remove(&s,&t);
    cout << "\nremoveu " << r;
    imprime(s);
    getch();
}
```

```
void insere(int y, celula **es, celula **et){  
    celula *nova = new celula;  
    nova -> conteudo = y;  
    nova -> seg = NULL;  
    if(*et == NULL) *et = *es = nova;  
    else{  
        (*et)->seg = nova;  
        *et = nova;  
    }  
}
```

- Remove

```
int remove(celula **es, celula **et){  
    celula *p;  
    int x;  
    p = *es;  
    x = p->conteudo;  
    *es = p->seg;  
    free(p);  
    if(*es == NULL) *et = NULL;  
    return x;  
}
```

# Imprime

```
void imprime(celula *s){  
    cout << "\nFila:";  
    while(s != NULL){  
        cout << " " << s->conteudo;  
        s = s->seg;  
    }  
}
```