



# Computação 2

## Aula II

### Filas e Pilhas

Prof<sup>a</sup>. Fabiany  
fabianyl@utfpr.edu.br

# Filas

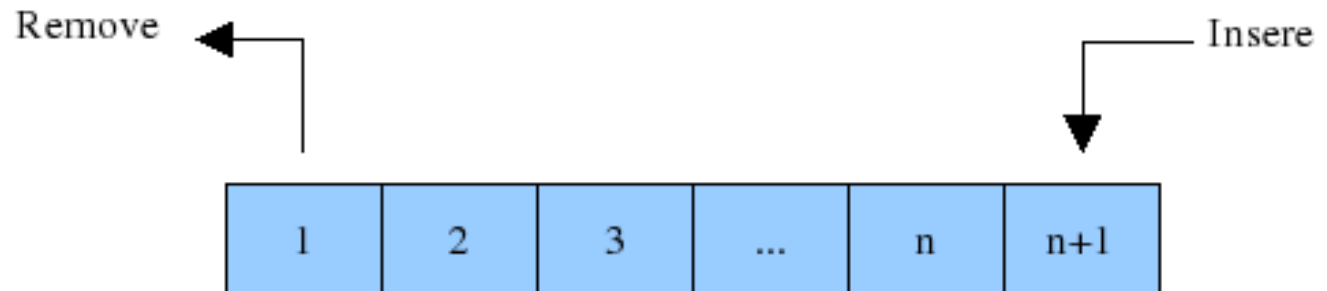
- Uma fila é simplesmente uma lista linear de informações;
- O acesso é feito na ordem primeiro a entrar, primeiro a sair, conhecido como FIFO (*First in First Out*).
- Ou seja, primeiro item colocado na fila é o primeiro a ser retirado, o segundo item colocado é o segundo a ser recuperado e assim por diante.

# Filas – Exemplos Aplicações

- Alocação de recursos para impressão de documento em um impressora;
- Atendimento de processos requisitados ao sistema operacional;
- Ordenação de encaminhamentos de pacotes em roteadores de rede;
- Buffer para gravação de dados em mídia;

# Filas

- Acesso FIFO (*First in First Out*).



# Filas

Inser(10)



primeiro  último 

Inser(20)



primeiro  último 

Inser(30)



primeiro  último 

Remove()



primeiro  último 

Inser(40)



primeiro  último 

Inser(50)



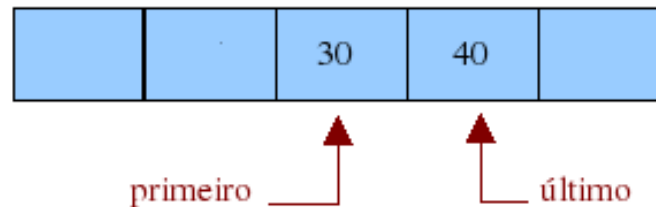
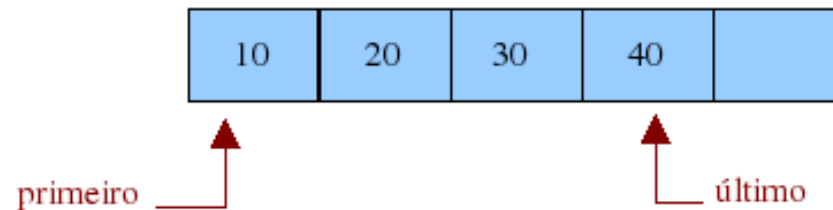
primeiro  último 

# Implementação Filas

Com vetor:

- Usar um tamanho fixo  $N$  de elementos;
- O processo de inserção e remoção em extremidades opostas fará com a fila “ande” pelo vetor;
- Exemplo: se inserimos os elementos 10, 20 30, 40 e depois retirarmos dois elementos, o começo da fila não estará mais nas posições iniciais do vetor.

# Filas - vetor

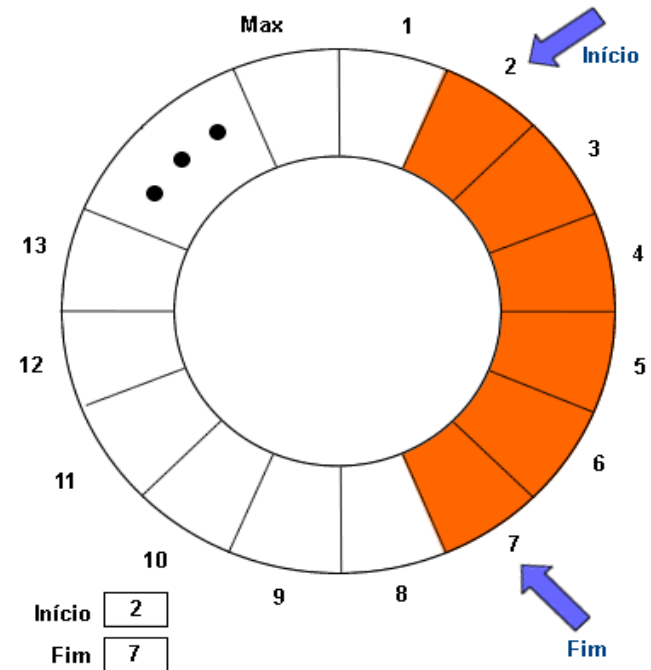
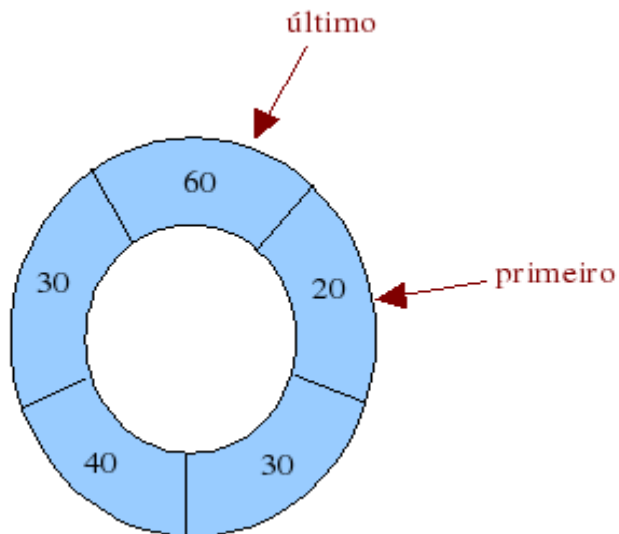


# Filas Circulares

- Observando a implementação de filas com vetor é possível notar que em determinado instante todos os elementos serão retirados e o vetor ficara livre, mas não será possível adicionar mais elementos;
- Para fazer um melhor aproveitamento dessa implementação, usa-se a fila circular. Se o ultimo elemento ocupa a ultima posição do vetor, inserimos os novos elementos a partir do inicio do vetor.



# Filas Circulares



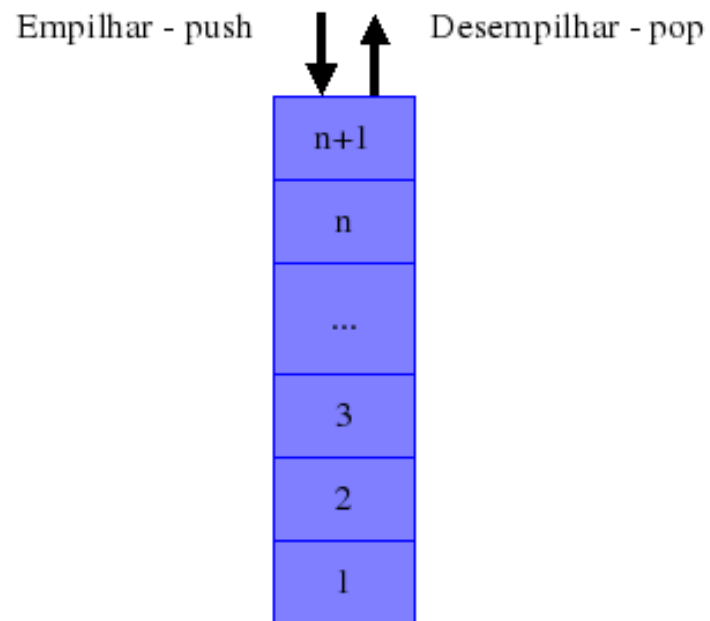
- Site com um exemplo iterativo de fila circular:

[http://www.ufpa.br/sampaio/curso\\_de\\_estdados\\_l/filas/filacirc\\_conceito.htm](http://www.ufpa.br/sampaio/curso_de_estdados_l/filas/filacirc_conceito.htm)

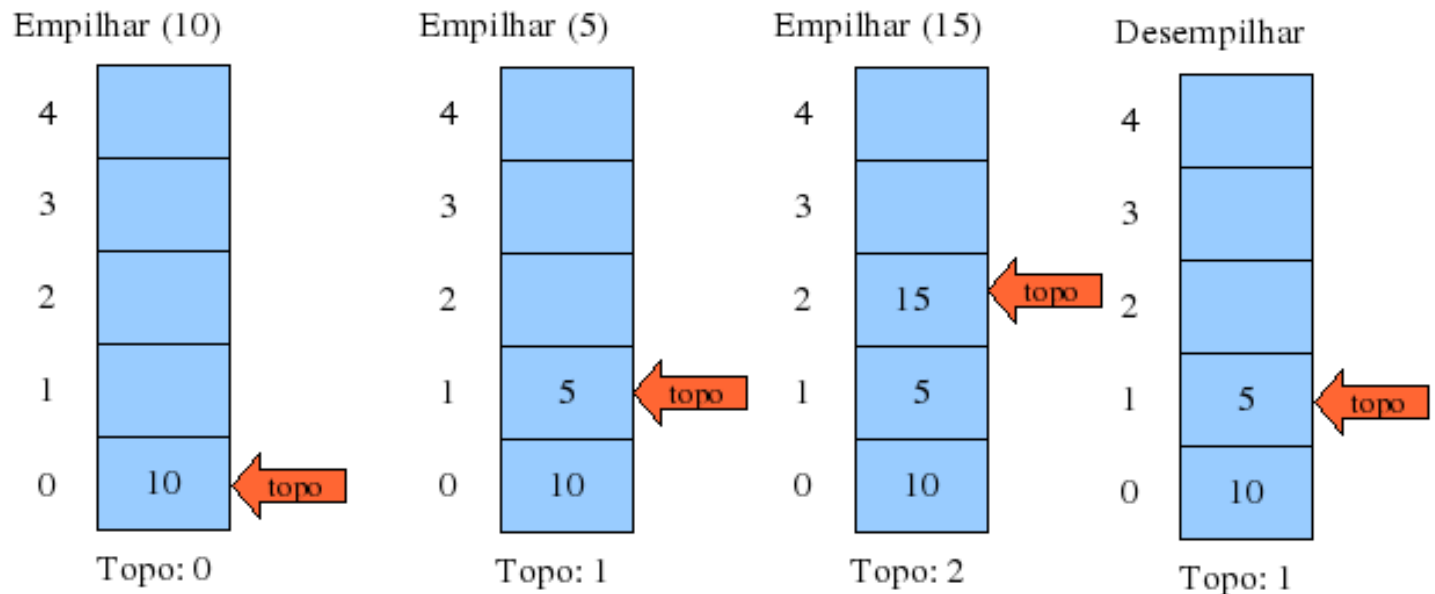
# Pilhas

- Uma pilha (ou *stack*) é o inverso de uma fila porque usa o acesso ultimo a entrar, primeiro a sair, também conhecido como LIFO (*Last In, Last Out*).
- Uma pilha é um estrutura de dados que admite a remoção de elementos (**desempilhar ou pop**) e a inserção de novos elementos (**empilhar ou push**).
- Entretanto, esta estrutura está sujeita à seguinte regra de operação: sempre que houver uma remoção, o elemento removido é o que está na estrutura há menos tempo.

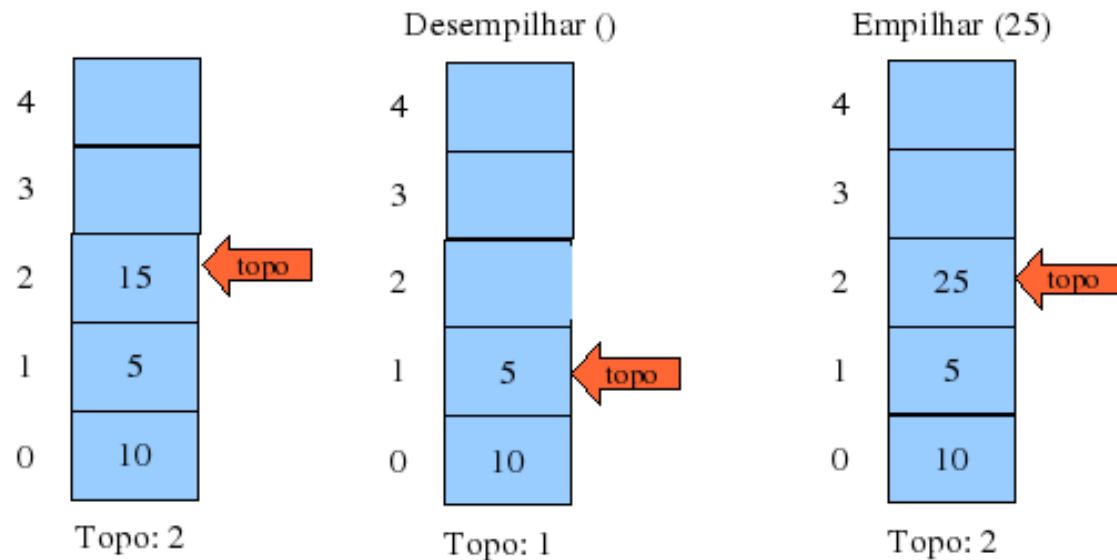
# Pilhas



# Pilhas - Exemplo de empilhamento (operação push)



# Exemplo de desempilhamento (operação pop)



# Exemplo - Pilha

```
#include <stdio.h>
#include <stdlib.h>

#define TAMANHO 5

// cabeçalho de função para empilhar número em um vetor
void empilha ( int i );

// cabeçalho de função para desempilhar número em um vetor
int desempilha ();

// ponteiro para topo (fim) de vetor
int *Topo;

// ponteiro para base (início) de vetor
int *Base;

// ponteiro para navegar sobre o vetor pilha via aritmética de ponteiros
int *Pont;

// vetor para pilha (primeiro número a entrar é ultimo a sair)
int Pilha[TAMANHO];
```

// função principal

int main()

{

int valor, cont = 0;

// Isto significa que o ponteiro Base recebe o endereço de início do vetor Pilha.

// (Note que não há & comercial para receber o endereço quando se trata de um vetor!)

// (Isto porque o próprio (nome do) vetor é internamente implementado como ponteiro...)

// (Isto é, o nome do vetor é um ponteiro)

Base = Pilha;

// O ponteiro Pont recebe o ponteiro Base

Pont = Base;

// O ponteiro Topo recebe o endereço de fim do vetor Pilha

Topo = Base + TAMANHO;

// O programa em si.

printf ("Programa para empilhar valores \n");

do

{

printf ( "Digite um valor: (-1 para parar) \n" );

scanf ( "%d", &valor );

if ( valor != -1 )

{

empilha (valor);

cont = cont + 1;

}

} while ( valor != -1 );

for ( int i = 0; i < cont; i++)

{

valor = desempilha();

printf("O %d o. valor desempilhado é %d \n", i+1, valor);

}

system("Pause");

return 0;

}

```
void empilha ( int i )
{
    // Se o ponteiro Pont estiver apontando já para o topo
    // então não se pode mais empilhar o número i

    if (Pont == Topo)
    {

        printf ("Pilha cheia! \n");
        // sai do programa
        exit (1);

    }
    else
    {

        // A "variável" apontada por Pont recebe o valor de i , i.e. a posição Pilha[0] recebe o valor de i.
        *Pont = i;

        // ponteiro Pont avança da sua posição inicial em 2 bytes (tamanho de um inteiro)
        // (i.e. Pont avança para seu endereço imediatamente superior)
        Pont++;

    }
}
```



```
int desempilha ( )
{
    // variável intermediária
    int valor;

    // Se o ponteiro Pont estiver num endereço maior ou igual que o do Base, isto é,
    // Se ele tiver endereço maior ou igual que início do Vetor Pilha...
    // então se extrai a informação em uma variável intermediária.
    if ( Pont >= Base )
    {
        // Ponteiro Pont aponta para seu endereço imediatamente inferior
        Pont--;

        // Valor recebe o valor da 'variável' apontado por Pont.
        valor = *Pont;
    }
    else
    {
        printf ("Pilha vazia! \n");
        exit (1);
    }
    return valor;
}
```

```
F:\Dev-CppPortable\App\devcpp\ProjectPilha.exe
Programa para empilhar valores
Digite um valor: <-1 para parar>
1
Digite um valor: <-1 para parar>
2
Digite um valor: <-1 para parar>
3
Digite um valor: <-1 para parar>
4
Digite um valor: <-1 para parar>
5
Digite um valor: <-1 para parar>
-1
0 1 o. valor desempilhado ú 5
0 2 o. valor desempilhado ú 4
0 3 o. valor desempilhado ú 3
0 4 o. valor desempilhado ú 2
0 5 o. valor desempilhado ú 1
Pressione qualquer tecla para continuar. . .
```