

LISTAS COM RESTRICÇÃO

Listas com restrição

- Para determinadas aplicações é imposto um critério que **restringe** a inserção e a retirada dos elementos que compõem um conjunto de dados.
- Duas restrições muito utilizadas são :

1. LIFO (*Last in First Out*): O **último** elemento que entra no conjunto é o **primeiro** a ser removido (PILHA)

Ex.: **No cotidiano** : Pilha de pratos, livros

Computação : Resolução de expressões, Recursividade, etc

2. FIFO (*First in First Out*): O **primeiro** elemento que entra no conjunto é o **primeiro** a ser removido

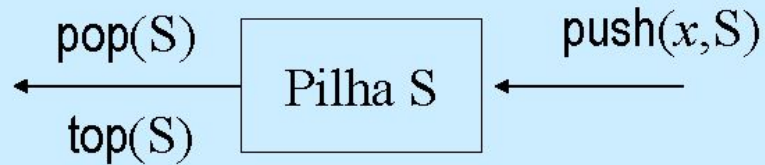
Ex.: **No cotidiano** : Fila de banco, Fila de impressão

Computação : Escalonamento de tarefas, etc

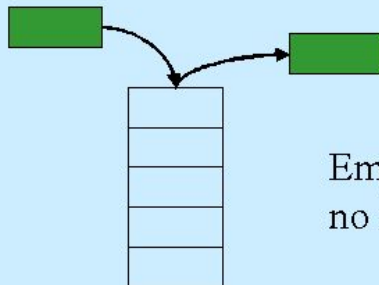
TAD PILHA

1 Pilhas

Pilhas são listas com a restrição de que as operações de **INSERÇÃO** e **DELEÇÃO** podem ser feitas em apenas uma posição, denominada o final da lista ou **TOP**.



Modelo geral de uma Pilha



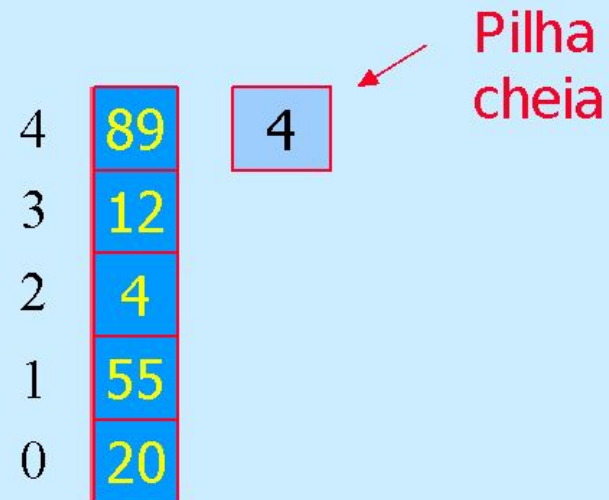
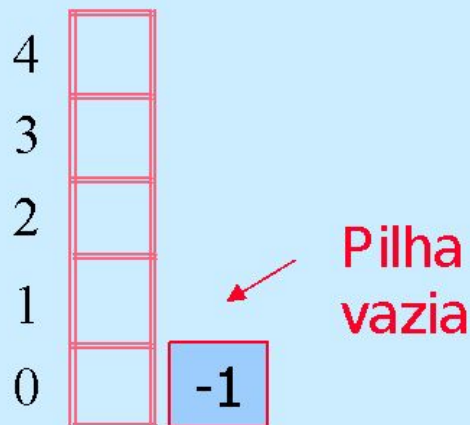
Empilhando e desempilhando
no modo LIFO - $O(1)$

1 Pilhas

- **Formas de implementação possíveis:**
 - Vetores
 - Listas encadeadas
- **Operações**
 - $\text{pop}(S)$ - Retira o elemento do topo (**desempilha**)
 - $\text{top}(S)$ - mostra o elemento do topo
 - $\text{push}(x, S)$ - Insere o elemento x na Pilha S (**empilha**)
- As **Pilhas** são estruturas essencialmente dinâmicas, podendo **crescer** e **decrescer** durante a execução de um programa.
- Quando se prevê o tamanho máximo o qual uma **Pilha** pode crescer, é preciso que não se defina um valor muito grande para que se use da melhor forma o recurso **Pilha**.

1 Pilha Sequencial

- Vetores possuem um espaço limitado para armazenamento de dados.
- Precisamos definir um espaço grande o suficiente para a nossa pilha.
- Precisamos de um indicador de qual elemento do vetor é o atual topo da pilha.

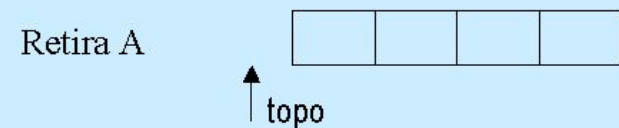
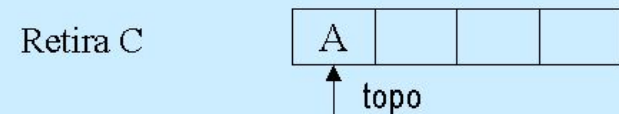
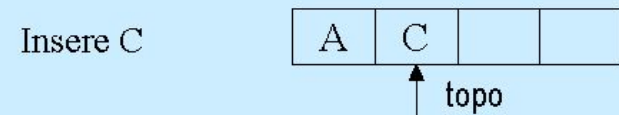
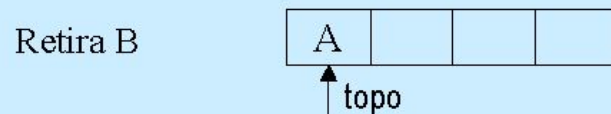
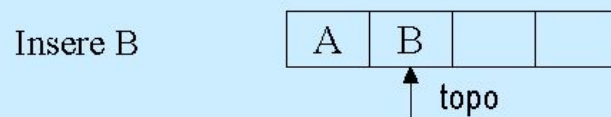
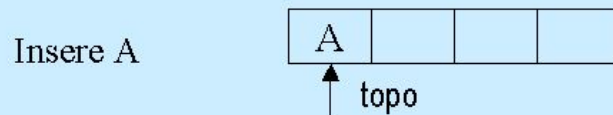
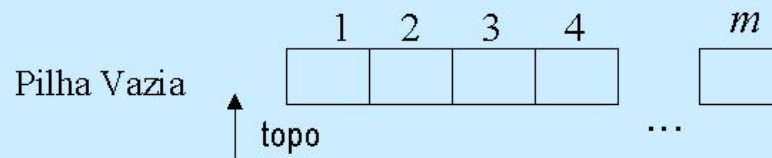


1 Pilhas

Aplicações

- Balanceamento de símbolos em equações
- Gerenciamento de memória

Funcionamento



Código Completo da TAD Pilha


```

public class TADPilha {
    private String nome;
    private int    topo;
    private int    capacidade;
    private int [] dados;

    //Construtor que cria uma Pilha do tamanho tam
    // -----
    public TADPilha (int tam, String nome){
        this.nome  = nome;
        capacidade = tam;
        topo       = -1;
        dados      = new int [capacidade];
    }

    // Método que imprime o índice do Topo
    // -----
    public void mostraIndiceTopo ()      {
        System.out.println("Topo da pilha "+nome + " = " + topo);
    }

    // Método que imprime o conteudo do Topo
    // -----
    public void mostraConteudoTopo () {
        if (topo == -1)
            System.out.print("Pilha " +
                             nome+"  VAZIA !!!");
        else
            System.out.print(
                "O conteudo da pilha " + nome + " = " + dados [topo]);
    }
}

```

```

// -----
public boolean cheia () {
    return (topo==capacidade - 1 );
}
// -----
public boolean empilha (int valor) {
    if (!cheia ()) {
        topo ++;
        dados [topo] = valor;
        return true;
    }
    return false;
}
// -----
public int desempilha () {
    if (topo != -1) {
        topo--;
        return dados [topo+ 1];
    }
    return -1;
}
// -----
public void imprimePilha () {
    if (topo != -1) {
        for (int i=topo; i >=0 ; i++) System.out.println(dados [i]);
    }
    else System.out.print("Pilha vazia !");
}

```

```
// -----  
public static void main () {  
  
    TADPilha p1=new TADPilha(2,"prato");  
    p1.empilha (5);  
    p1.empilha (4);  
    p1.empilha (3);  
    p1.desempilha();  
    p1.empilha (2);  
    p1.imprimePilha(); //Somente 5 e 4  
}  
  
// -----  
}
```