



**UNIVERSIDADE FEDERAL DO PARÁ  
CAMPUS UNIVERSITÁRIO DE CASTANHAL  
FACULDADE DE COMPUTAÇÃO**

**CÓDIGOS ESTRUTURAS DE DADOS PILHAS E FILAS**

**Pilhas contiguas: Exemplo de um vetor de 5 posições:**

//Classe da estrutura pilha contigua

```
package estudoPilha;

public class Pilha {

    // pilha contigua
    int Topo;
    int[] Elem = new int[5];
}
```

//Classe dos métodos de manipulação da Pilha e o método principal estão no mesmo arquivo

package estudoPilha;

```
public class metodoPilha {
    public static int tamanho = 5;

    // iniciar a Pilha
    public static void inicia(Pilha P) {
        P.Topo = -1;
    }
}
```

```
//verificar se a PILHA está vazia
public static boolean vazia(Pilha P) {
    if (P.Topo == -1) {
        return true;
    } else {
        return false;
    }
}
```

```
//verificar se a PILHA está cheia
public static boolean overflow(Pilha P) {
    if (P.Topo == tamanho-1) {
        return true;
    } else {
        return false;
    }
}
```

```
// inserir elementos na PILHA
public static void push(Pilha P, int V) {
    if (P.Topo == tamanho-1) {
        System.out.println("Pilha Cheia!");
    } else {
        P.Topo++;
        P.Elem[P.Topo] = V;
    }
}
```

```
//remover elementos da PILHA
public static void pop(Pilha P) {
    if (P.Topo == -1) {
        System.out.println("Pilha Vazia!");
    } else {
        P.Topo--;
    }
}
```

```
//Ler o elemento da Pilha
public static int elemento(Pilha P) {
    return (P.Elem[P.Topo]);
}

public static void LerPilha(Pilha P) {
    System.out.println("O ultimo elemento
da pilha é: "+ P.Elem[P.Topo]);
}
```

// método principal - **o que será executado?**

```
public static void main(String args[]) {  
    Pilha P1 = new Pilha();  
  
    System.out.println("Iniciando a Pilha");  
    inicia(P1);  
  
    System.out.println("Inserindo itens na pilha...");  
    push(P1, 1);  
    push(P1, 2);  
    push(P1, 3);  
    push(P1, 4);  
    push(P1, 5);  
  
    if (overflow(P1)) {  
        System.out.println("Pilha cheia, não se pode inserir mais itens");  
    } else {  
        push(P1, 6);  
    }  
  
    System.out.println("Acessa o topo da pilha");  
    System.out.println(elemento(P1));  
    System.out.println(elemento(P1));  
  
    System.out.println("Desempilhando a Pilha");  
    pop(P1);  
    System.out.println(elemento(P1));  
  
    System.out.println("Desempilhando a Pilha novamente");  
    pop(P1);  
    System.out.println(elemento(P1));  
  
    pop(P1);  
    pop(P1);  
    pop(P1);  
  
    if (vazia(P1)) {  
        System.out.println("Pilha vazia, não se pode retirar itens");  
    } else {  
        pop(P1);  
    }  
}
```

## Pilhas Encadeadas:

//Classe da estrutura pilha encadeada (nó)

```
package estudoPilha;
public class noPilha {
    // PILHAS ENCADEADAS
    public int valor;
    public noPilha proximo;
}
```

//Classe dos métodos da estrutura pilha encadeada

package estudoPilha;

public class metodonoPilha {

private noPilha inicio;

metodonoPilha() {  
 //inicializar a Pilha  
 inicio = null;  
 }

//verificar se a PILHA esta vazia

```
    public boolean vazia() {
        if (inicio == null) {
            return true;
        } else {
            return false;
        }
    }
```

//Ler o elemento da Pilha

```
    public void elemento() {
        if (vazia()){
            System.out.println("Pilha Vazia - utilizado o metodo vazia()");
        }else{
            System.out.println(inicio.valor);
        }
    }
}
```

```
// inserir elementos na PILHA
    public void push(int V) {
        noPilha q = new noPilha();
        q.valor = V;
        q.proximo = inicio;
        inicio = q;
    }
```

```
//remover elementos da PILHA
    public void pop() {
        if (inicio == null) {
            System.out.println("Pilha vazia");
        } else {
            inicio = inicio.proximo;
        }
    }
```

//Classe principal, do método main: **o que será executado?**

```
package estudoPilha;
```

```
public class principalNoPilha {
```

```
    public static void main (String args[]){  
        metodonoPilha P1 = new metodonoPilha();
```

```
        P1.push(30);  
        P1.elemento();  
        P1.push(8);
```

```
        P1.push(13);  
        P1.push(40);  
        P1.push(80);
```

```
        P1.push(33);  
        P1.push(43);  
        P1.pop();
```

```
        P1.elemento();
```

```
        P1.push(8);  
        P1.elemento();  
        P1.push(1);  
        P1.elemento();
```

```
        P1.pop();  
        P1.elemento();
```

```
        P1.pop();  
        P1.elemento();  
        P1.pop();  
        P1.elemento();
```

```
        P1.pop();  
        P1.elemento();
```

```
        P1.pop();  
        P1.elemento();
```

```
        P1.push(18);  
        P1.elemento();
```

```
        P1.pop();  
        P1.elemento();
```

```
    }  
}
```

## Filas Encadeadas:

//Classe da estrutura Fila (nó)

package estudoFila;

```
public class noFila {  
    // FILAS ENCADEADAS  
  
    public int valor;  
    public noFila proximo;  
  
    // Ponteiros de controle  
    public noFila primeiro;  
    public noFila ultimo;  
}
```

//Classe dos métodos da estrutura Fila

package estudoFila;

```
public class metodonoFila {
```

```
    noFila F = new noFila();
```

```
    // private noFila inicio;
```

```
    metodonoFila() {
```

```
        F.primeiro = null;
```

```
        F.ultimo = null;
```

```
    }
```

//verificar se a FILA esta vazia

```
    public boolean vazia() {  
        if (F.primeiro == null) {  
            return true;  
        } else {  
            return false;  
        }  
    }
```

//Ler o elemento da Fila

```
    public void elemento() {  
        if (vazia() == true){  
            System.out.println("Metodo Elemento  
- Fila vazia...");  
        } else {  
            System.out.println(F.primeiro.valor);  
        }  
    }  
}
```

// inserir elementos na FILA

```
    public void inserir(int V) {  
        noFila q = new noFila();  
        q.valor = V;  
        q.proximo = null;  
        if (F.primeiro == null) { // fila vazia  
            F.primeiro = q;  
            F.ultimo = q;  
        } else {  
            F.ultimo.proximo = q;  
            F.ultimo = q;  
        }  
    }
```

//remover elementos na FILA

```
    public void remover() {  
  
        if (F.primeiro == null) { // fila vazia  
            System.out.println("Fila vazia");  
        } else {  
            F.primeiro = F.primeiro.proximo;  
            if (F.primeiro == null) {  
                F.ultimo = null;  
            }  
        }  
    }
```

//Classe principal, do método main: **o que será executado?**

```
package estudoFila;
```

```
public class principalnoFila {
```

```
    public static void main(String args[]) {  
        metodonoFila F1 = new metodonoFila();
```

```
        F1.inserir(3);  
        F1.elemento();
```

```
        F1.inserir(8);  
        F1.elemento();
```

```
        F1.inserir(1);  
        F1.elemento();
```

```
        F1.remover();  
        F1.elemento();
```

```
        F1.remover();  
        F1.elemento();
```

```
        F1.remover();  
        F1.elemento();
```

```
        F1.remover();  
        F1.elemento();  
        F1.remover();  
        F1.elemento();
```

```
    }  
}
```