

PRACTICA SOBRE METODOS DE ORDENACION CON LA ESTRUCTURA STRATEGY

CLASE MAIN:

```
package ConBusqueda;

import java.util.Scanner;

/**
 *
 * @author MIGUEL EDUARDO
 */
public class Main {

    public static void main(String[] args) {

        Scanner leer = new Scanner(System.in);

        int resp1, resp2;

        System.out.println("***** Ingresá los Datos *****");
        System.out.println();
        System.out.println();

        System.out.println("***** ¿Qué realizarás? ***** ");
        System.out.println("1      Ordenación ");
        System.out.println("2      Búsqueda ");
        System.out.println();
        resp1 = leer.nextInt();

        if(resp1==1){
            System.out.println("***** Seleccioná un Método para ordenar el vector ***** ");
            System.out.println("1      Burbuja ");
            System.out.println("2      Inserción ");
            System.out.println("3      Selección ");
            System.out.println();
            resp2 = leer.nextInt();

            System.out.println(" Ingresá el Número de valores del Arreglo ");
            int [] vector = new int [leer.nextInt()];
            System.out.println("OK, ahora ingresa los valores ");
        }
    }
}
```



```
for(int i= 0;i<vector.length;i++)
{
    System.out.println("Ingrese el Valor " + (i+1) + " = ");
    vector [i] = leer.nextInt();
}

switch(resp2){

    case 1:
        MetBurbuja b = new MetBurbuja();
        vector = b.ordenar(vector);
        System.out.println();
        System.out.println("Los Valores Ordenados del Arreglo son: ");
        for(int i=0;i<vector.length;i++)
        {
            System.out.print(vector[i]+ " - ");
        }
        break;

    case 2:

        MetInsercion I = new MetInsercion ();
        vector = I.ordenar(vector);
        System.out.println();
        System.out.println("Los Valores Ordenados del Vector son: ");
        for(int i=0;i<vector.length;i++)
        {
            System.out.print(vector [i]+ " - ");
        }
        break;

    case 3:
        MetSeleccion S = new MetSeleccion ();
        vector = S.ordenar(vector);
        System.out.println();
        System.out.println("Los Valores Ordenados del Vector son: ");
        for(int i=0;i<vector.length;i++)
        {
            System.out.print(vector [i]+ " - ");
        }

        break;
}
}
```



```
else{
    System.out.println("***** Selecciona un Tipo de Busqueda");
    **** ");
    System.out.println("1) Binaria ");
    System.out.println("2) Secuencial ");
    System.out.println();
    resp2 = leer.nextInt();

    System.out.println(" Ingresa el Numero de valores del Arreglo");
}

int [] vector = new int [leer.nextInt()];
System.out.println("OK, ahora ingresa los valores ");
for(int i= 0;i<vector.length;i++)
{
    System.out.println("Ingrese el Valor " + (i+1) + " = ");
    vector [i] =leer.nextInt();
}

System.out.println("Bien, Ahora ingresa el numero que deseas
Buscar ");
int nbuscado = leer.nextInt();

switch (resp2){

    case 1:
        BusBinaria b= new BusBinaria();
        nbuscado = b.busca(vector, resp2);
        break;

    case 2:
        BusSecuencial s= new BusSecuencial();
        nbuscado = s.busca(vector, resp2);
        break;
    }

    if(nbuscado>-1)
    {
        System.out.println("El numero esta ubicado en el lugar
numero: "+nbuscado);
    }else{
        System.out.println("El numero no Existe ");
    }
}
}
```



CLASE ORDENA:

```
package ConBusqueda;

abstract class Ordena {

    abstract int [] ordenar (int [] vector);

}
```

CLASE METBURBUJA:

```
package Strategy;

public class MetBurbuja extends Metodos{

    int [] ordenar (int[] vector){

        for(int i=0; i<vector.length; i++)
        {
            for(int j=i; j<vector.length; j++)
                if(!(vector[i]<vector[j])){
                    int aux= vector[i];
                    vector[i]=vector[j];
                    vector[j]=aux;
                }
        }

        return vector;
    }
}
```

CLASE METINSERCIÓN:

```
package Strategy;

public class MetInsercion extends Metodos {

    int [] ordenar (int[] vector) {
        for (int i=1; i<vector.length; i++) {
            int aux = vector[i];
            int j;
            for (j=i-1; j>=0 && vector[j]>aux; j--)

```



```
        vector[j+1] = vector[j];
        vector[j+1] = aux;
    }

    return vector;
}
```

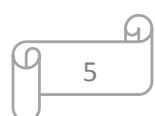
CLASE METSELECCION:

```
package Strategy;

public class MetSeleccion extends Metodos {

    int [] ordenar (int[] vector)
{
    for (int i = 0; i < vector.length - 1; i++)
    {
        int min = i;
        for (int j = i + 1; j < vector.length; j++)
        {
            if (vector[j] < vector[min])
            {
                min = j;
            }
        }
        if (i != min)
        {
            int aux= vector[i];
            vector[i] = vector[min];
            vector[min] = aux;
        }
    }

    return vector;
}
}
```



CLASE BUSCAR:

```
package ConBusqueda;

/**
 *
 * @author MIGUEL EDUARDO
 */
abstract class Buscar {

    abstract int busca (int [] vector, int num);

}
```

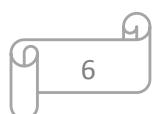
CLASE BUSBINARIA:

```
package ConBusqueda;

/**
 *
 * @author MIGUEL EDUARDO
 */
public class BusBinaria extends Buscar {

    int busca (int[] vector, int num) {

        int inicio = 0;
        int fin = vector.length - 1;
        int pos;
        int[] f = new int[1];
        vector = this.busca(vector);
        while (inicio <= fin) {
            pos = (inicio+fin) / 2;
            if ( vector[pos] == num ) {
                f[0] = pos;
                return pos;
            }
            else if ( vector[pos] < num ) {
                inicio = pos+1;
            } else {
                fin = pos-1;
            }
        }
        int[] a = new int[1];
        a[0] = -1;
    }
}
```



```
return -1;
}
int[] busca (int[] n) {
    int x;
    for (int i = 1; i < n.length; i++) {
        for (int k = n.length- 1; k >= i; k--) {
            if(n[k] < n[k-1]){
                x = n[k];
                n[k] = n[k-1];
                n[k-1] = x;
            }
        }
    }
    return n;
}
}
```

CLASE BUSSECUENCIAL:

```
package ConBusqueda;

/**
 *
 * @author MIGUEL EDUARDO
 */
public class BusSecuencial extends Buscar{
    int busca(int[] vector, int num) {

        int[] f = new int[1];
        vector = this.busca(vector);
        int flag = 0;
        for(int x = 0 ; x < vector.length ; x++){
            if(vector[x]==num) {
                flag = 1;
                f[0] = x;
                return 2;
            }
        }
        if(flag==0){
            f[0] = -1;
        }return 1;
    }
    int[] busca(int[] n) {
        int x;
        for (int i = 1; i < n.length; i++) {
            for (int k = n.length- 1; k >= i; k--) {
```



```
if(n[k] < n[k-1]) {  
    x = n[k];  
    n[k] = n[k-1];  
    n[k-1] = x;  
}  
}  
return n;  
}  
}
```

DIAGRAMA DE CLASES

