



PRACTICA 8

Facultad de ciencias de la computación
Ing. En ciencias de la computación

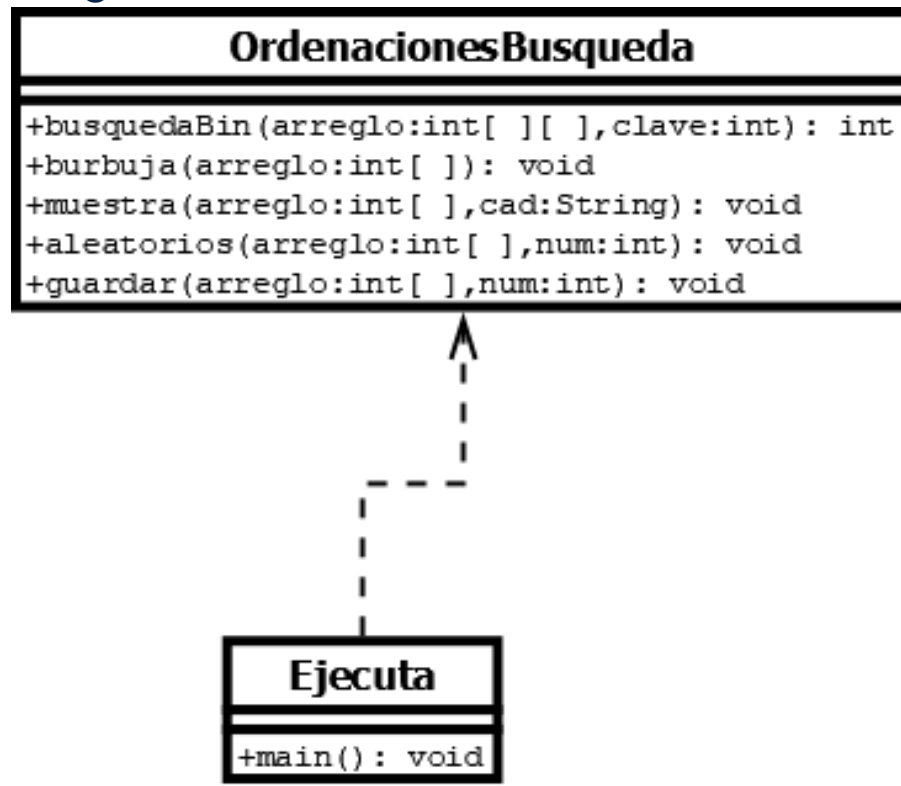


IRVYN XICALE CABRERA
201963582

Introducción

Recordar lo previamente aprendido en programación 1 con los tipos de ordenaciones y búsquedas.

Diagramas ULM



Código

OrdenacionesBusqueda.java

```
import java.io.IOException;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class OrdenacionsBusqueda {

    public static void main(String[] args)throws IOException {
        String n, datoEntrada;
        int op=0, op1=0, op2=0, op3=0, op4=0, op5=0, op6=0, op7=0, op8=0, op9=0, op10=0;
        int op11=0, op12=0, op13=0, op14=0, op15=0;
        InputStreamReader entrada = new InputStreamReader(System.in);
        BufferedReader Entrada = new BufferedReader(entrada);
        System.out.println("-/-/-/-/Metodos de Ordenamiento y Busqueda/-/-/-/");
        System.out.println("Ingrese el numero de elementos que tendra el arreglo: ");
        datoEntrada=Entrada.readLine();
        int num=Integer.parseInt(datoEntrada);
        int arreglo1 []=new int[num];
```

```

Ordenacion_Busqueda objeto=new Ordenacion_Busqueda();
do{
    System.out.println("----- Menu -----");
    System.out.println("¿Que metodo de ordenamiento deseas usar?");
    System.out.println("Opcion 1: Burbuja");
    System.out.println("Opcion 2: Burbuja Mejorado");
    System.out.println("Opcion 3: Seleccion");
    System.out.println("Opcion 4: Shell");
    System.out.println("Opcion 5: QuickSort");
    System.out.println("Opcion 6: Salir del programa");

    datoEntrada=Entrada.readLine();
    op=Integer.parseInt(datoEntrada);
switch(op){
    case 1:
        do{
            System.out.println("¿De que forma desea ordenarlo?");
            System.out.println("Opcion 1: De forma Ascendente | De menor a mayor |");
            System.out.println("Opcion 2: De forma Descendente | De mayor a menor |");

```

```
System.out.println("Opcion 3: Regresar al menu");
    datoEntrada=Entrada.readLine();
    op1=Integer.parseInt(datoEntrada);
    switch(op1){
        case 1: objeto.aleatorios(arreglo1, num);
    objeto.burbuja(arreglo1);
        objeto.muestra(arreglo1, "Ordenado por Burbuja de forma Ascendente");
    objeto.guardar(arreglo1, num);
        do{
            System.out.println("¿De que forma quiere buscar un elemento");
            System.out.println("1. Busqueda Binaria");
            System.out.println("2.Busqueda Secuencial");
            System.out.println("3. Regresar al menú principal");
            datoEntrada=Entrada.readLine();
            op2=Integer.parseInt(datoEntrada);
        switch(op2){
            case 1:
                System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
                String num1=Entrada.readLine();
```

```

    int num2=Integer.parseInt(num1);
    int indice=objeto.busquedaBin(arreglo1,num2);
    if (indice!=-1){
        System.out.println("El numero "+num2+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
    }else{
        System.out.println("El numero "+num2+" no se encuentra en el arreglo");
    }
    break;
case 2:
    System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
    String num3=Entrada.readLine();
    int num4=Integer.parseInt(num3);
    int indice1=objeto.busquedaLineal(arreglo1,num4);
    if(indice1!=-1){
        System.out.println("El numero "+num4+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
    }else{
        System.out.println("El numero "+num4+" no se encuentra en el arreglo");
    }
    break;

```

```

case 3:
    System.out.println("----- Regresando al menú principal -----");
    break;
default: System.out.println("La opción no existe");
    break;
}
}while(op2!=3);
break;
case 2: objeto.aleatorios(arreglo1, num);
    objeto.burbuja2(arreglo1);
    objeto.muestra(arreglo1, "Ordenado por Burbuja de forma Descendente");
    objeto.guardar(arreglo1, num);
    do{
        System.out.println("¿De que forma quiere buscar un elemento");
        System.out.println("1. Busqueda Binaria");
        System.out.println("2.Busqueda Secuencial");
        System.out.println("3. Regresar al menú principal");
        datoEntrada=Entrada.readLine();
        op3=Integer.parseInt(datoEntrada);
    }

```

```

switch(op3){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num5=Entrada.readLine();
        int num6=Integer.parseInt(num5);
        int indice=objeto.busquedaBin(arreglo1,num6);
        if (indice!=-1){
            System.out.println("El numero "+num6+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num6+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num7=Entrada.readLine();
        int num8=Integer.parseInt(num7);
        int indice1=objeto.busquedaLineal(arreglo1,num8);
        if(indice1!=-1){
            System.out.println("El numero "+num8+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
        }
    }
}

```



```

        }else{
            System.out.println("El numero "+num8+" no se encuentra en el arreglo");
        }
        break;
    case 3: System.out.println("Regresando al menu principal");
        break;
    default: System.out.println("La opción no existe");
        break;
    }
}while(op3!=3);
break;
case 3: System.out.println("Regresando al menu principal");
break;
default: System.out.println("La opcion no existe");
break;
}
}while(op1!=3);
case 2:
do{

```

```
System.out.println("¿De que forma desea ordenarlo?");
System.out.println("1.De forma Ascendente (De menor a mayor)");
System.out.println("2.De forma Descendente (De mayor a menor)");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op4=Integer.parseInt(datoEntrada);
switch(op4){
    case 1: objeto.aleatorios(arreglo1, num);
    objeto.burbujaMejorado(arreglo1);
    objeto.muestra(arreglo1, "Ordenado por Burbuja Mejorado de forma Ascendente");
    objeto.guardar(arreglo1, num);
    do{
        System.out.println("¿De que forma quiere buscar un elemento");
        System.out.println("1. Busqueda Binaria");
        System.out.println("2.Busqueda Secuencial");
        System.out.println("3. Regresar al menú principal");
        datoEntrada=Entrada.readLine();
        op5=Integer.parseInt(datoEntrada);
        switch(op5){
```

case 1:

```
System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
String num9=Entrada.readLine();
int num10=Integer.parseInt(num9);
int indice=objeto.busquedaBin(arreglo1,num10);
if (indice!=-1){
    System.out.println("El numero "+num10+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
}else{
    System.out.println("El numero "+num10+" no se encuentra en el arreglo");
}
break;
```

case 2:

```
System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
String num11=Entrada.readLine();
int num12=Integer.parseInt(num11);
int indice1=objeto.busquedaLineal(arreglo1,num12);
if(indice1!=-1){
    System.out.println("El numero "+num12+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
}else{
```

```

        System.out.println("El numero "+num12+" no se encuentra en el arreglo");
    }
    break;
    case 3: System.out.println("----- Regresando al menú principal -----");
    break;
    default: System.out.println("La opción no existe");
    break;
}
}while(op5!=3);
break;
case 2: objeto.aleatorios(arreglo1, num);
objeto.burbujaMejorado2(arreglo1);
objeto.muestra(arreglo1, "Ordenado por Burbuja Mejorado de forma Descendente");
objeto.guardar(arreglo1, num);
do{
    System.out.println("¿De que forma quiere buscar un elemento");
    System.out.println("1. Búsqueda Binaria");
    System.out.println("2.Búsqueda Secuencial");
    System.out.println("3. Regresar al menú principal");

```

```
datoEntrada=Entrada.readLine();
op6=Integer.parseInt(datoEntrada);
switch(op6){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num13=Entrada.readLine();
        int num14=Integer.parseInt(num13);
        int indice=objeto.busquedaBin(arreglo1,num14);
        if (indice!=-1){
            System.out.println("El numero "+num14+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num14+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num15=Entrada.readLine();
        int num16=Integer.parseInt(num15);
        int indice1=objeto.busquedaLineal(arreglo1,num16);
```

```

        if(indice1!=-1){
            System.out.println("El numero "+num16+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num16+" no se encuentra en el arreglo");
        }
        break;
    case 3: System.out.println("Regresando al menu principal");
        break;
    default: System.out.println("La opción no existe");
        break;
    }
    }while(op6!=3);
    break;
    case 3: System.out.println("Regresando al menu principal");
        break;
    default: System.out.println("La opcion no existe");
        break;
    }
    }while(op4!=3);

```

```
case 3:
do{
System.out.println("¿De que forma desea ordenarlo?");
System.out.println("1.De forma Ascendente (De menor a mayor)");
System.out.println("2.De forma Descendente (De mayor a menor)");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op7=Integer.parseInt(datoEntrada);
switch(op7){
    case 1: objeto.aleatorios(arreglo1, num);
    objeto.seleccion(arreglo1);
    objeto.muestra(arreglo1, "Ordenado por Seleccion de forma Ascendente");
    objeto.guardar(arreglo1, num);
    do{
        System.out.println("¿De que forma quiere buscar un elemento");
        System.out.println("1. Busqueda Binaria");
        System.out.println("2.Busqueda Secuencial");
        System.out.println("3. Regresar al menú principal");
        datoEntrada=Entrada.readLine();
```

```

op8=Integer.parseInt(datoEntrada);
switch(op8){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num17=Entrada.readLine();
        int num18=Integer.parseInt(num17);
        int indice=objeto.busquedaBin(arreglo1,num18);
        if (indice!=-1){
            System.out.println("El numero "+num18+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num18+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num19=Entrada.readLine();
        int num20=Integer.parseInt(num19);
        int indice1=objeto.busquedaLineal(arreglo1,num20);
        if(indice1!=-1){

```



```

        System.out.println("El numero "+num20+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
    }else{
        System.out.println("El numero "+num20+" no se encuentra en el arreglo");
    }
    break;
case 3: System.out.println("----- Regresando al menú principal -----");
    break;
default: System.out.println("La opción no existe");
    break;
}
}while(op8!=3);
break;
case 2: objeto.aleatorios(arreglo1, num);
objeto.seleccion2(arreglo1);
objeto.muestra(arreglo1, "Ordenado por Seleccion de forma Descendente");
objeto.guardar(arreglo1, num);
do{
    System.out.println("¿De que forma quiere buscar un elemento");
    System.out.println("1. Busqueda Binaria");

```

```
System.out.println("2.Busqueda Secuencial");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op9=Integer.parseInt(datoEntrada);
switch(op9){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num21=Entrada.readLine();
        int num22=Integer.parseInt(num21);
        int indice=objeto.busquedaBin(arreglo1,num22);
        if (indice!=-1){
            System.out.println("El numero "+num22+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num22+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num23=Entrada.readLine();
```

```

int num24=Integer.parseInt(num23);
int indice1=objeto.busquedaLineal(arreglo1,num24);
if(indice1!=-1){
    System.out.println("El numero "+num24+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
}else{
    System.out.println("El numero "+num24+" no se encuentra en el arreglo");
}
break;
case 3: System.out.println("Regresando al menu principal");
break;
default: System.out.println("La opción no existe");
break;
}
}while(op9!=3);
break;
case 3: System.out.println("Regresando al menu principal");
break;
default: System.out.println("La opcion no existe");
break;

```

}

```
}while(op7!=3);
```

case 4:

do{

```
System.out.println("¿De que forma desea ordenarlo?");
```

```
System.out.println("1.De forma Ascendente (De menor a mayor)");
```

```
System.out.println("2.De forma Descendente (De mayor a menor)");
```

```
System.out.println("3. Regresar al menú principal");
```

```
datoEntrada=Entrada.readLine();
```

```
op10=Integer.parseInt(datoEntrada);
```

```
switch(op10){
```

```
case 1: objeto.aleatorios(arreglo1, num);
```

```
objeto.shell(arreglo1);
```

```
objeto.muestra(arreglo1, "Ordenado por Shell de forma Ascendente");
```

```
objeto.guardar(arreglo1, num);
```

do{

```
System.out.println("¿De que forma quiere buscar un elemento");
```

```
System.out.println("1. Busqueda Binaria");
```

```
System.out.println("2.Busqueda Secuencial");
```

```
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op11=Integer.parseInt(datoEntrada);
switch(op11){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num25=Entrada.readLine();
        int num26=Integer.parseInt(num25);
        int indice=objeto.busquedaBin(arreglo1,num26);
        if (indice!=-1){
            System.out.println("El numero "+num26+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num26+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num27=Entrada.readLine();
        int num28=Integer.parseInt(num27);
```

```

int indice1=objeto.busquedaLineal(arreglo1,num28);
if(indice1!=-1){
    System.out.println("El numero "+num28+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
}else{
    System.out.println("El numero "+num28+" no se encuentra en el arreglo");
}
break;
case 3: System.out.println("----- Regresando al menú principal -----");
break;
default: System.out.println("La opción no existe");
break;
}
}while(op11!=3);
break;
case 2: objeto.aleatorios(arreglo1, num);
objeto.shell2(arreglo1);
objeto.muestra(arreglo1, "Ordenado por Shell de forma Descendente");
objeto.guardar(arreglo1, num);
do{

```

```
System.out.println("¿De que forma quiere buscar un elemento");
System.out.println("1. Búsqueda Binaria");
System.out.println("2. Búsqueda Secuencial");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op12=Integer.parseInt(datoEntrada);
switch(op12){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num29=Entrada.readLine();
        int num30=Integer.parseInt(num29);
        int indice=objeto.búsquedaBin(arreglo1,num30);
        if (indice!=-1){
            System.out.println("El numero "+num30+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num30+" no se encuentra en el arreglo");
        }
        break;
    case 2:
```

```

        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num31=Entrada.readLine();
        int num32=Integer.parseInt(num31);
        int indice1=objeto.busquedaLineal(arreglo1,num32);
        if(indice1!=-1){
            System.out.println("El numero "+num32+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num32+" no se encuentra en el arreglo");
        }
        break;
    case 3: System.out.println("Regresando al menu principal");
        break;
    default: System.out.println("La opción no existe");
        break;
    }
}while(op12!=3);
break;
case 3: System.out.println("Regresando al menu principal");
break;

```



```

        default: System.out.println("La opcion no existe");
        break;
    }
}while(op10!=3);
case 5:
do{
System.out.println("¿De que forma desea ordenarlo?");
System.out.println("1.De forma Ascendente (De menor a mayor)");
System.out.println("2.De forma Descendente (De mayor a menor)");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op13=Integer.parseInt(datoEntrada);
switch(op13){
    case 1: objeto.aleatorios(arreglo1, num);
    objeto.quickSort(arreglo1,0,0);
    objeto.muestra(arreglo1, "Ordenado por QuickSort de forma Ascendente");
    objeto.guardar(arreglo1, num);
    do{
        System.out.println("¿De que forma quiere buscar un elemento");
    }
}
}

```

```
System.out.println("1. Búsqueda Binaria");
System.out.println("2. Búsqueda Secuencial");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op14=Integer.parseInt(datoEntrada);
switch(op14){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num33=Entrada.readLine();
        int num34=Integer.parseInt(num33);
        int indice=objeto.búsquedaBin(arreglo1,num34);
        if (indice!=-1){
            System.out.println("El numero "+num34+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num34+" no se encuentra en el arreglo");
        }
        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
```

```

String num35=Entrada.readLine();
int num36=Integer.parseInt(num35);
int indice1=objeto.busquedaLineal(arreglo1,num36);
if(indice1!=-1){
    System.out.println("El numero "+num36+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
}else{
    System.out.println("El numero "+num36+" no se encuentra en el arreglo");
}
break;
case 3: System.out.println("----- Regresando al menú principal -----");
break;
default: System.out.println("La opción no existe");
break;
}
}while(op14!=3);
break;
case 2: objeto.aleatorios(arreglo1, num);
objeto.quickSort2(arreglo1,0,0);
objeto.muestra(arreglo1, "Ordenado por QuickSort de forma Descendente");

```

```

objeto.guardar(arreglo1, num);
do{
System.out.println("¿De que forma quiere buscar un elemento");
System.out.println("1. Busqueda Binaria");
System.out.println("2.Busqueda Secuencial");
System.out.println("3. Regresar al menú principal");
datoEntrada=Entrada.readLine();
op15=Integer.parseInt(datoEntrada);
switch(op15){
    case 1:
        System.out.println("Ingrese el numero del elemento a buscar dentro el arreglo: ");
        String num37=Entrada.readLine();
        int num38=Integer.parseInt(num37);
        int indice=objeto.busquedaBin(arreglo1,num38);
        if (indice!=-1){
            System.out.println("El numero "+num38+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num38+" no se encuentra en el arreglo");
        }
    }
}

```

```

        break;
    case 2:
        System.out.println("Ingrese el numero del elemento a buscar dentro del arreglo :");
        String num39=Entrada.readLine();
        int num40=Integer.parseInt(num39);
        int indice1=objeto.busquedaLineal(arreglo1,num40);
        if(indice1!=-1){
            System.out.println("El numero "+num40+" se encuentra en posicion " + (indice1+1)+ " en el arreglo");
        }else{
            System.out.println("El numero "+num40+" no se encuentra en el arreglo");
        }
        break;
    case 3: System.out.println("Regresando al menu principal");
        break;
    default: System.out.println("La opción no existe");
        break;
    }
}while(op15!=3);
break;

```

```
        case 3: System.out.println("Regresando al menu principal");
        break;
        default: System.out.println("La opcion no existe");
        break;
    }
}while(op13!=3);
case 6: System.out.println("<<<<< Gracias por usar nuestro sistema >>>>>");
break;
default: System.out.println("La opción no existe");
}
}while(op!=6);
}

}
```

Ejecuta.java

```
import java.io.*;  
import java.util.Scanner;
```

```
public class Ejecuta{  
    public static void main(String[] args) throws IOException  
    {  
  
        int n;  
        Scanner lec=new Scanner(System.in);  
        System.out.println("dar numero de elementos en el arreglo");  
        n=lec.nextInt();  
        int arreglo1[]=new int[n];  
        OrdenacionesBusqueda objeto=new OrdenacionesBusqueda();  
        objeto.aleatorios(arreglo1,n);  
        objeto.burbuja(arreglo1);  
        objeto.muestra(arreglo1,"Ordenado por Burbuja ");  
        objeto.guardar(arreglo1,n);  
        System.out.println("dar numero de elemento a buscar en el arreglo");
```

```
n=lec.nextInt();
int indice=objeto.busquedaBin(arreglo1,n);
if (indice!=-1)
{
    System.out.println("El numero "+n+" se encuentra en posicion " + (indice+1)+ " en el arreglo");
}
else
{
    System.out.println("El numero "+n+" no se encuentra en el arreglo");
}

}

}
```



```
Go Run Terminal Help Metodos_Ordenamiento.java - MenuOrdenacion - Visual Studio Code

PROBLEMS 43 OUTPUT DEBUG CONSOLE TERMINAL 1: Java Process Console v + [] [x] v x

-/-/-/-/Metodos de Ordenamiento y Busqueda/-/-/-/
Ingrese el numero de elementos que tendra el arreglo:
5
----- Menu -----
¿Que metodo de ordenamiento deseas usar?
Opcion 1: Burbuja
Opcion 2: Burbuja Mejorado
Opcion 3: Seleccion
Opcion 4: Shell
Opcion 5: QuickSort
Opcion 6: Salir del programa
1
¿De que forma desea ordenarlo?
Opcion 1: De forma Ascendente |De menor a mayor|
Opcion 2: De forma Descendente |De mayor a menor|
Opcion 3: Regresar al menu
2
numero 45
numero 18
numero 36
numero 95
numero 97
Ordenado por Burbuja de forma Descendente97 95 45 36 18

¿De que forma quiere buscar un elemento
1. Busqueda Binaria
2.Busqueda Secuencial
3. Regresar al menú principal
2
Ingrese el numero del elemento a buscar dentro del arreglo :
18
Opcion 5: QuickSort
Opcion 6: Salir del programa
1
¿De que forma desea ordenarlo?
Opcion 1: De forma Ascendente |De menor a mayor|
Opcion 2: De forma Descendente |De mayor a menor|
```

```
Go Run Terminal Help Metodos_Ordenamiento.java - MenuOrdenacion - Visual Studio Code

PROBLEMS 43 OUTPUT DEBUG CONSOLE TERMINAL 1: Java Process Console v + [] ✕ x

2
Ingrese el numero del elemento a buscar dentro del arreglo :
18
Opcion 5: QuickSort
Opcion 6: Salir del programa
1
¿De que forma desea ordenarlo?
Opcion 1: De forma Ascendente |De menor a mayor|
Opcion 2: De forma Descendente |De mayor a menor|
Opcion 3: Regresar al menu
2
numero 45
numero 18
numero 36
numero 95
numero 97
Ordenado por Burbuja de forma Descendente97 95 45 36 18

¿De que forma quiere buscar un elemento
1. Busqueda Binaria
2.Busqueda Secuencial
3. Regresar al menú principal
2
Ingrese el numero del elemento a buscar dentro del arreglo :
18
Si encontró:18
El numero 18 se encuentra en posición 19 en el arreglo
¿De que forma quiere buscar un elemento
1. Busqueda Binaria
2.Busqueda Secuencial
3. Regresar al menú principal
█
```

Conclusión

Logramos reforzar los conocimientos aprendidos sobre los métodos de ordenación y búsquedas.

Hoja de firmas

RÚBRICA HOJA DE EVALUACIÓN DE PRÁCTICAS

MATRICULA: 201963582

FECHA: 24/01/2021

NOMBRE: Xicale Cabrera Irvyn

No. PRACTICA: 1

INDIVIDUAL (x) COLABORATIVA ()

CRITERIOS	Deficiente	Suficiente	Bueno	Excelente	Calificación Obtenida	
	1-5.9	6-7.9	8-9	9.1-10	%	Puntos
CONOCIMIENTO TEÓRICO 20%	Conocimiento deficiente de los fundamentos teóricos de POO y no puede aplicarlos en el diseño de clases.	Conocimiento confuso de los fundamentos teóricos de POO y el diseño de las clases y relaciones es incompleto.	Conocimiento claro de los fundamentos teóricos POO pero requiere mejorar el modelado de las clases, sus métodos y sus relaciones entre clases.	Dominio del Conocimiento de los fundamentos teóricos POO y puede aplicarlos de forma completa en el modelado de las clases, métodos y todas las relaciones entre clase.		
EJECUCIÓN DE LA PRÁCTICA 30%	No puede realizar la práctica ya que desconoce el entorno de trabajo y desarrollo de la práctica en lenguaje UML y Java.	Realiza la práctica de forma incompleta ya que desconoce el entorno de trabajo del lenguaje UML y Java	Realiza la práctica pero requiere mejorar en el manejo del entorno de trabajo del lenguaje de programación(sintaxis y semántica)	Realiza la práctica de forma correcta y completa, demuestra dominio del entorno de trabajo del lenguaje de programación (sintaxis y semántica).		
SOLUCIÓN DE LA PRÁCTICA 40%	No puede generar las soluciones o programas a los problemas planteados ya que no posee el dominio teórico y práctico del modelado y el lenguaje de programación.	Propone soluciones confusas o programas incompletos a los problemas planteados, ya que carece del dominio del modelado y lenguaje de programación	Genera soluciones con poca profundidad y los programas no están orientados de acuerdo a los problemas solicitados, por lo cual no tiene un dominio profundo de la temática y del Lenguaje.	Genera soluciones con profundidad y los programas son correctos de acuerdo a los problemas planteados, por lo cual demuestra un dominio de la temática y del Lenguaje de Programación.		

ACTITUD DE APRENDER Y COLABORAR EN EQUIPO DE TRABAJO 10%	No posee una actitud proactiva para un aprendizaje autónomo y no le gusta participar y trabajar en equipo.	Posee una actitud propositiva para un aprendizaje autónomo, participa pero no le gusta trabajar en equipo.	Posee una actitud propositiva logrando un aprendizaje autónomo, colaborativo, le gusta trabajar en equipo pero requiere mejorar su participación y portaciones de forma profunda.	Posee una actitud proactiva logrando un aprendizaje autónomo, participa con propuestas concretas y profundas, le gusta trabajar en equipo y asume su responsabilidad dentro para lograr éxito del equipo		
Total						

Firma del Alumn@

Vo. Bo. Docente

