

Universidad Mariano Gálvez

Facultad de Ingeniería en Sistemas de Información y Ciencias de la Computación

Boca del Monte

Programación III

Sección: "B"



## **METODOS DE ORDENAMIENTO**

# Métodos de ordenamiento

## 1. SELECCIÓN

```
Output - MetodosOrdenamiento (run) X
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
1
1.Selection sort (selección)
Ingrese la cantidad de números a ordenar: 5
Ingrese el número 1: 2
Ingrese el número 2: 9
Ingrese el número 3: 1
Ingrese el número 4: 3
Ingrese el número 5: 7

Array antes del ordenamiento:
2 9 1 3 7

Array después del ordenamiento:
1 2 3 7 9
BUILD SUCCESSFUL (total time: 15 seconds)
```

## 2. Burbuja

```
Output - MetodosOrdenamiento (run) X
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
2
2.Bubble sort (burbuja)
Ingrese los números separados por espacios:
2 3 4 -2 10
Array original:
2 3 4 -2 10

Array ordenado:
-2 2 3 4 10
BUILD SUCCESSFUL (total time: 9 seconds)
```

### 3. Inserción

```
>
Output - MetodosOrdenamiento (run) X
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
3
3.Insertion sort (inserción)

Ingrese la cantidad de elementos: 4
Ingrese los elementos:
10 -2 5 9

Array original:
10 -2 5 9

Array ordenado:
-2 5 9 10
BUILD SUCCESSFUL (total time: 17 seconds)
|
```

### 4. Combinación

```
>
Output - MetodosOrdenamiento (run) X
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
4
4.Merge sort (combinación)

Ingrese la cantidad de elementos a ordenar: 5
Ingrese los elementos:
10 -2 8 2 1

Arreglo ordenado:
-2 1 2 8 10 BUILD SUCCESSFUL (total time: 20 seconds)
|
```

## 5. Rápida

```
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
5
5.Quick sort (rápida)

Ingrese la cantidad de elementos a ordenar: 4
Ingrese los elementos a ordenar:
10 0 3 1000

Elementos ordenados:
0 3 10 1000 BUILD SUCCESSFUL (total time: 13 seconds)
```

## 6. Montón

S

```
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
6
6.Heap sort (montón)

Ingrese la cantidad de números a ordenar: 7
Ingrese los números a ordenar:
-2 5 10 4 3 8 9
Números ordenados:
-2 3 4 5 8 9 10 BUILD SUCCESSFUL (total time: 19 seconds)
```

## 7. Conteo

```
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
7
7.Counting sort (conteo)

Ingrese la cantidad de números a ordenar: 8
Ingrese los números a ordenar:
1000 3 2 -1 9 6 8 13
Arreglo ordenado: [-1, 2, 3, 6, 8, 9, 13, 1000]
BUILD SUCCESSFUL (total time: 33 seconds)
```

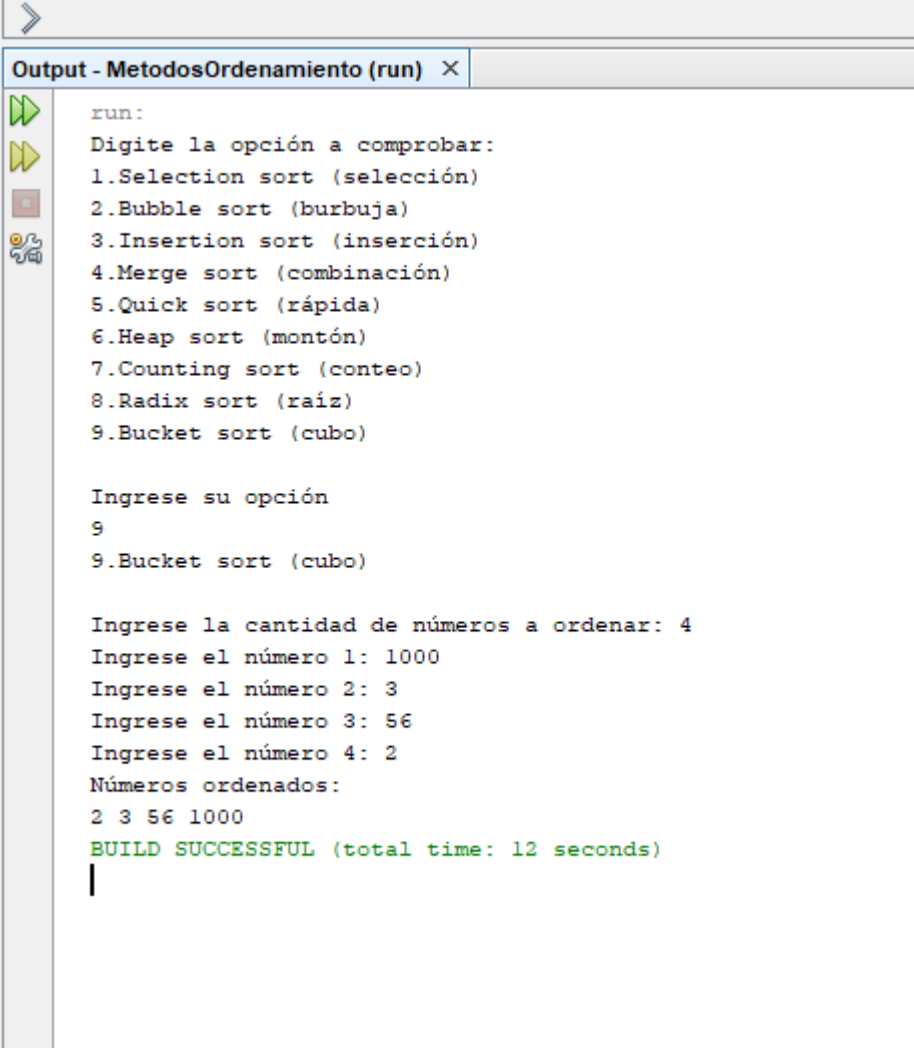
## 8. Raíz

```
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

Ingrese su opción
8
8.Radix sort (raíz)

Ingrese la cantidad de números a ordenar:
4
Ingrese los números a ordenar:
2
6
1
100
Números ordenados:
1 2 6 100 BUILD SUCCESSFUL (total time: 19 seconds)
```

## 9. Cubo



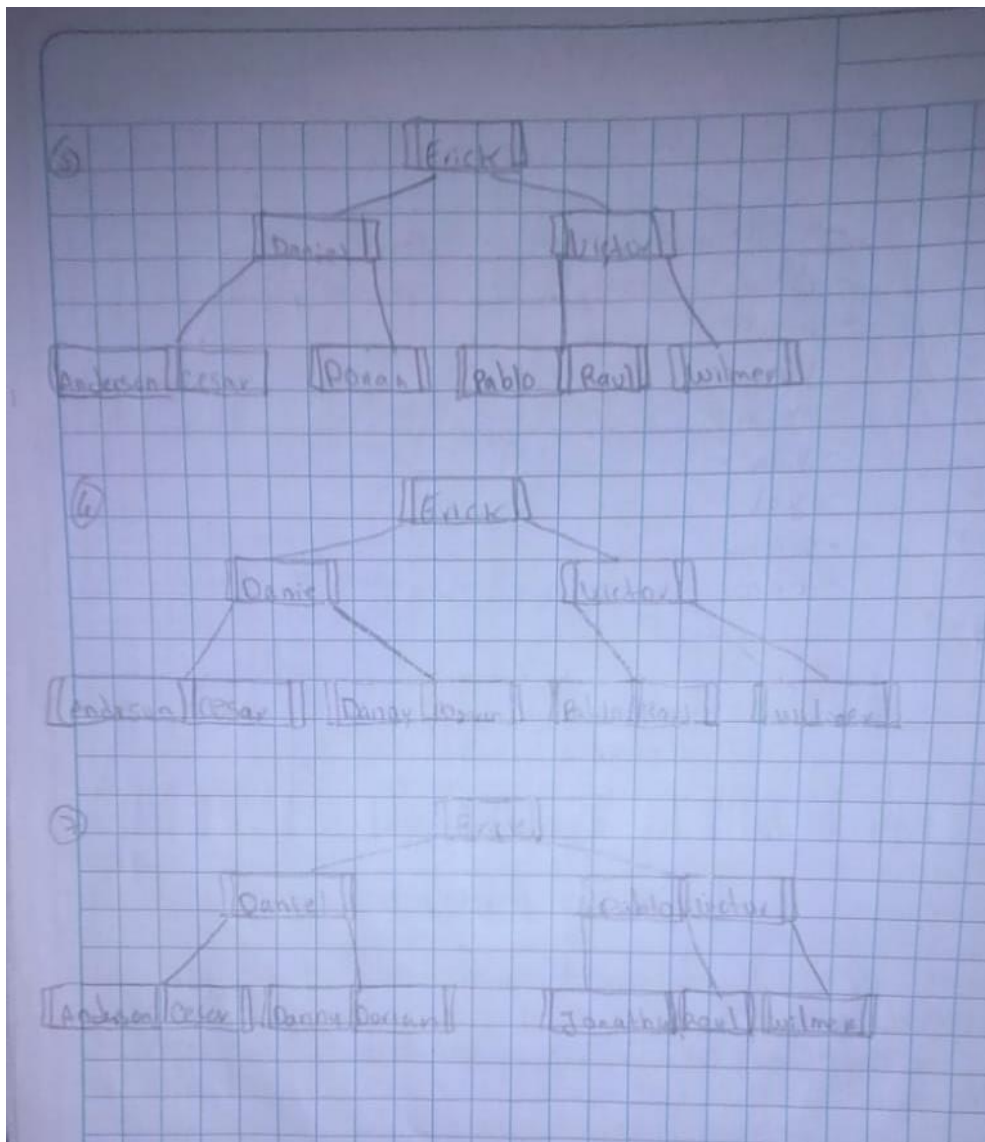
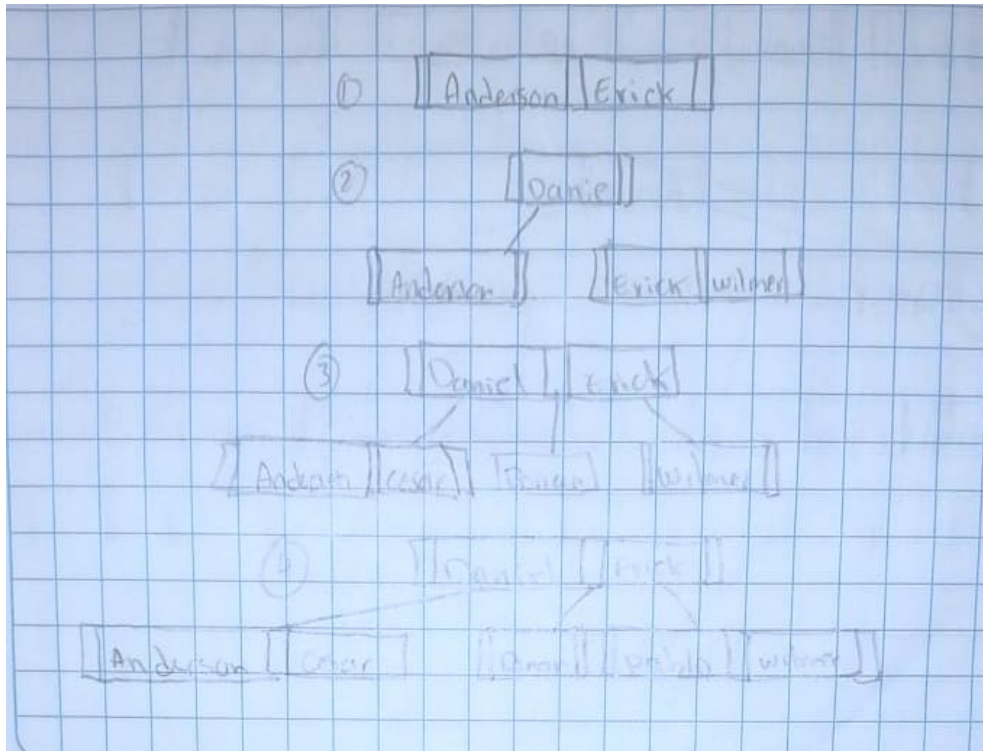
The screenshot shows a Java IDE window titled "Output - MetodosOrdenamiento (run) X". The output text is as follows:

```
run:
Digite la opción a comprobar:
1.Selection sort (selección)
2.Bubble sort (burbuja)
3.Insertion sort (inserción)
4.Merge sort (combinación)
5.Quick sort (rápida)
6.Heap sort (montón)
7.Counting sort (conteo)
8.Radix sort (raíz)
9.Bucket sort (cubo)

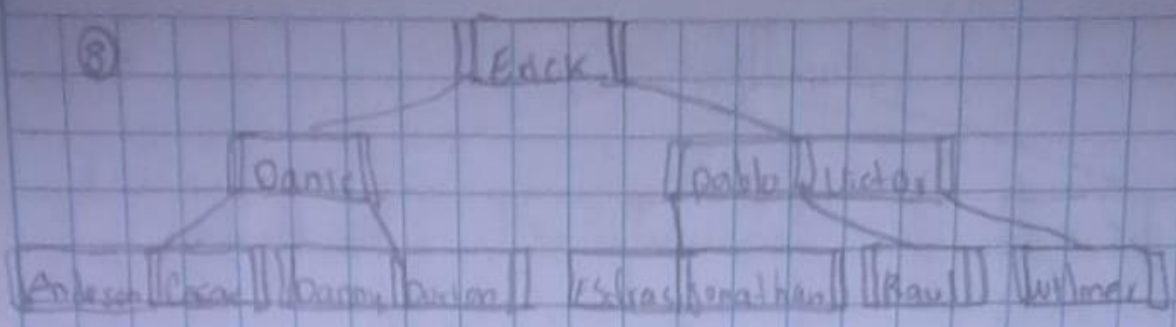
Ingrese su opción
9
9.Bucket sort (cubo)

Ingrese la cantidad de números a ordenar: 4
Ingrese el número 1: 1000
Ingrese el número 2: 3
Ingrese el número 3: 56
Ingrese el número 4: 2
Números ordenados:
2 3 56 1000
BUILD SUCCESSFUL (total time: 12 seconds)
```

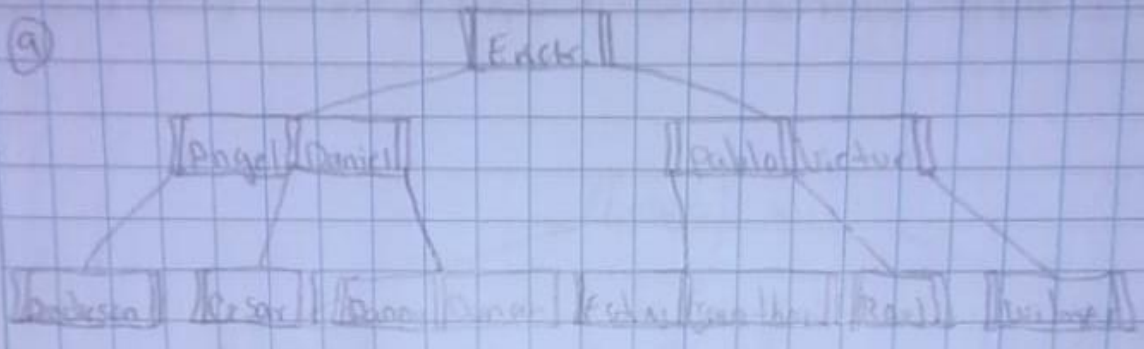
# EJERCICIO REALIZADO EN CLASE



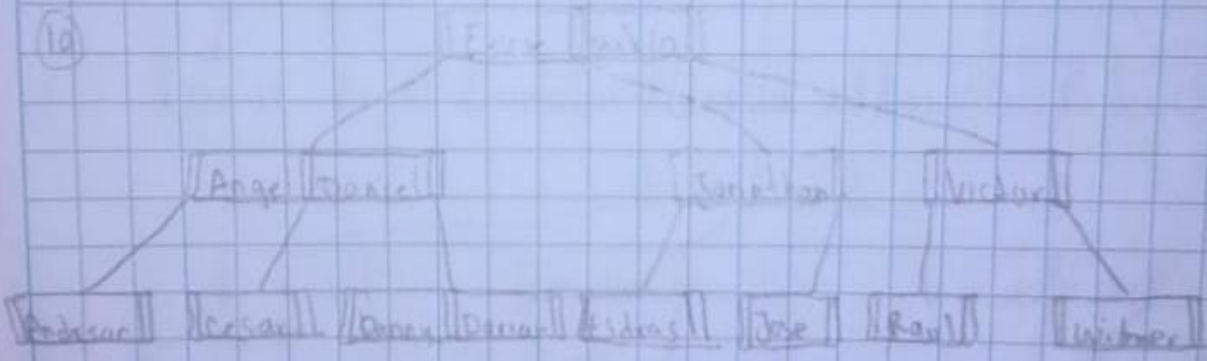
8



9

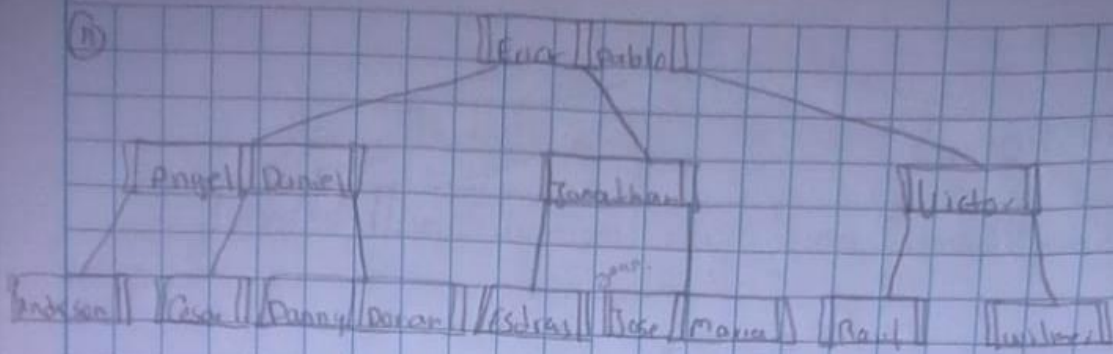


10





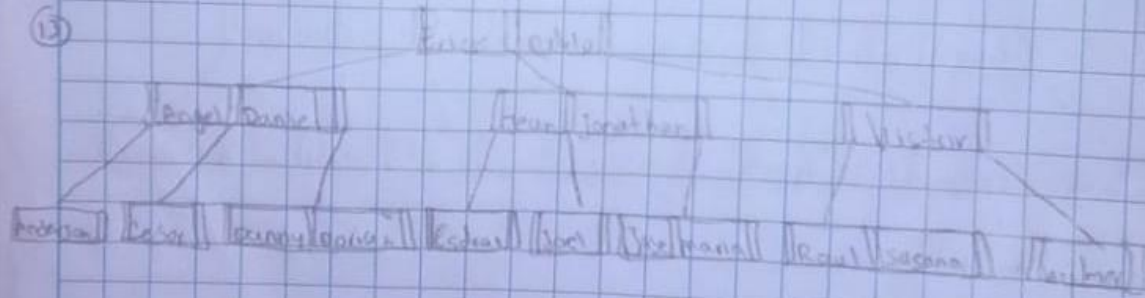
11



12



13



## RESULTADO FINAL

