

```
run:
ingrese la cantidad de elementos:
2
ingrese los elementos:
-1
3
seleccione el metodo de ordenamiento:
1. selection sort
2. bubble sort
3. insertion sort
4. merge sort
5. quick sort
6. heap sort
7. counting sort
8. Radix sort
9. Bucket sort
4
arreglo ordenado: [-1, 3]
BUILD SUCCESSFUL (total time: 15 seconds)
|
```

```
run:
ingrese la cantidad de elementos:
1
ingrese los elementos:
3
seleccione el metodo de ordenamiento:
1. selection sort
2. bubble sort
3. insertion sort
4. merge sort
5. quick sort
6. heap sort
7. counting sort
8. Radix sort
9. Bucket sort
1
arreglo ordenado: [3]
BUILD SUCCESSFUL (total time: 8 seconds)
|
```

```
run:
ingrese la cantidad de elementos:
6
ingrese los elementos:
-1
-2
-3
-5
78
8
seleccione el metodo de ordenamiento:
1. selection sort
2. bubble sort
3. insertion sort
4. merge sort
5. quick sort
6. heap sort
7. counting sort
8. Radix sort
9. Bucket sort
2
arreglo ordenado: [-5, -3, -2, -1, 8, 78]
BUILD SUCCESSFUL (total time: 12 seconds)
|
```

```
tarea_2 > bucketSort >
Output - tarea_2 (run) x Notifications
run:
ingrese la cantidad de elementos:
4
ingrese los elementos:
5
8
99
1
seleccione el metodo de ordenamiento:
1. selection sort
2. bubble sort
3. insertion sort
4. merge sort
5. quick sort
6. heap sort
7. counting sort
8. Radix sort
9. Bucket sort
3
arreglo ordenado: [1, 5, 8, 99]
BUILD SUCCESSFUL (total time: 14 seconds)
```

```
tarea_2 > bucketSort >
Output - tarea_2 (run) x Notifications
run:
ingrese la cantidad de elementos:
4
ingrese los elementos:
3
4
5
6
seleccione el metodo de ordenamiento:
1. selection sort
2. bubble sort
3. insertion sort
4. merge sort
5. quick sort
6. heap sort
7. counting sort
8. Radix sort
9. Bucket sort
8
arreglo ordenado: [3, 4, 5, 6]
BUILD SUCCESSFUL (total time: 7 seconds)
```