

# **11. Métodos de Ordenamiento**

**Padilla Herrera Carlos Ignacio**

**Folio: 794DR02**

**Escuela de Código PILARES  
Java Entre Semana G0224**

**Fecha: 12 de junio de 2024**

# 11. Métodos de Ordenamiento

Carlos Ignacio Padilla Herrera

12 de junio de 2024

## 11. Métodos de Ordenamiento

**Objetivo:** Verificar el dominio teórico del concepto de colas en Java mediante ejercicios para desarrollar su código y preguntas de respuesta múltiple.

**Indicaciones:** Realiza los programas que se te solicitan, una vez realizado sube el archivo de tu actividad.

## Actividades

### Programa 1. (3 puntos)

Se tiene el siguiente arreglo con 5 elementos, desarrolla el procedimiento completo para ordenarlo por el método de ordenamiento de inserción:

```
1 public class OrdenamientoPorInsercion {
2
3     public static void main(String[] args) {
4         int[] arreglo = {5, 2, 4, 6, 1, 3};
5         ordenamientoPorInsercion(arreglo);
6
7         System.out.print("Arreglo ordenado: ");
8         for (int num : arreglo) {
9             System.out.print(num + " ");
10        }
11    }
12
13    public static void ordenamientoPorInsercion(int[] arreglo) {
14        for (int i = 1; i < arreglo.length; i++) {
15            int clave = arreglo[i];
16            int j = i - 1;
17
18            while (j >= 0 && arreglo[j] > clave) {
19                arreglo[j + 1] = arreglo[j];
20                j = j - 1;
21            }
22            arreglo[j + 1] = clave;
23        }
24    }
25 }
```

Listing 1: Código base del programa OrdenamientoPorInsercion.java

## Programa 2. (3 puntos)

Completa el siguiente código de tal forma que el resultado obtenido sea el que se muestra en la imagen:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this
   license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4  */
5  package metododeordenamiento;
6
7  public class MetodoDeOrdenamiento {
8
9      public void imprimeArreglo(String mensaje, int[] arreglo) {
10         System.out.print(mensaje);
11         for (int num : arreglo) {
12             System.out.print(num + " ");
13         }
14         System.out.println();
15     }
16
17     public void saltoLinea() {
18         System.out.println("\n");
19     }
20
21     public static void main(String args[]) {
22         int[] arreglo = {11, 23, 9, 20};
23
24         MetodoDeOrdenamiento objOrdena = new MetodoDeOrdenamiento();
25
26         objOrdena.imprimeArreglo("Arreglo inicial: ", arreglo);
27         objOrdena.saltoLinea();
28
29         for (int i = 0; i < arreglo.length; i++) {
30             System.out.println("i = " + i);
31             for (int j = 0; j < arreglo.length - 1; j++) {
32                 objOrdena.imprimeArreglo(" Arreglo comparando: ", arreglo);
33                 System.out.println(" <----- j = " + j + " -- " + arreglo[j] + " > " + arreglo[j + 1]
34                     + "?");
35
36                 if (arreglo[j] > arreglo[j + 1]) {
37                     System.out.println(" ----- Intercambia " + arreglo[j] + " con " + arreglo[j +
38                         1]);
39                     // Intercambio
40                     int temp = arreglo[j];
41                     arreglo[j] = arreglo[j + 1];
42                     arreglo[j + 1] = temp;
43
44                     objOrdena.imprimeArreglo(" Arreglo modificado: ", arreglo);
45                     objOrdena.saltoLinea();
46                 }
47             }
48         }
49
50         objOrdena.saltoLinea();
51         objOrdena.imprimeArreglo("Arreglo final: ", arreglo);
52     }
53 }
```

Listing 2: Código base del programa MetodoDeOrdenamiento.java

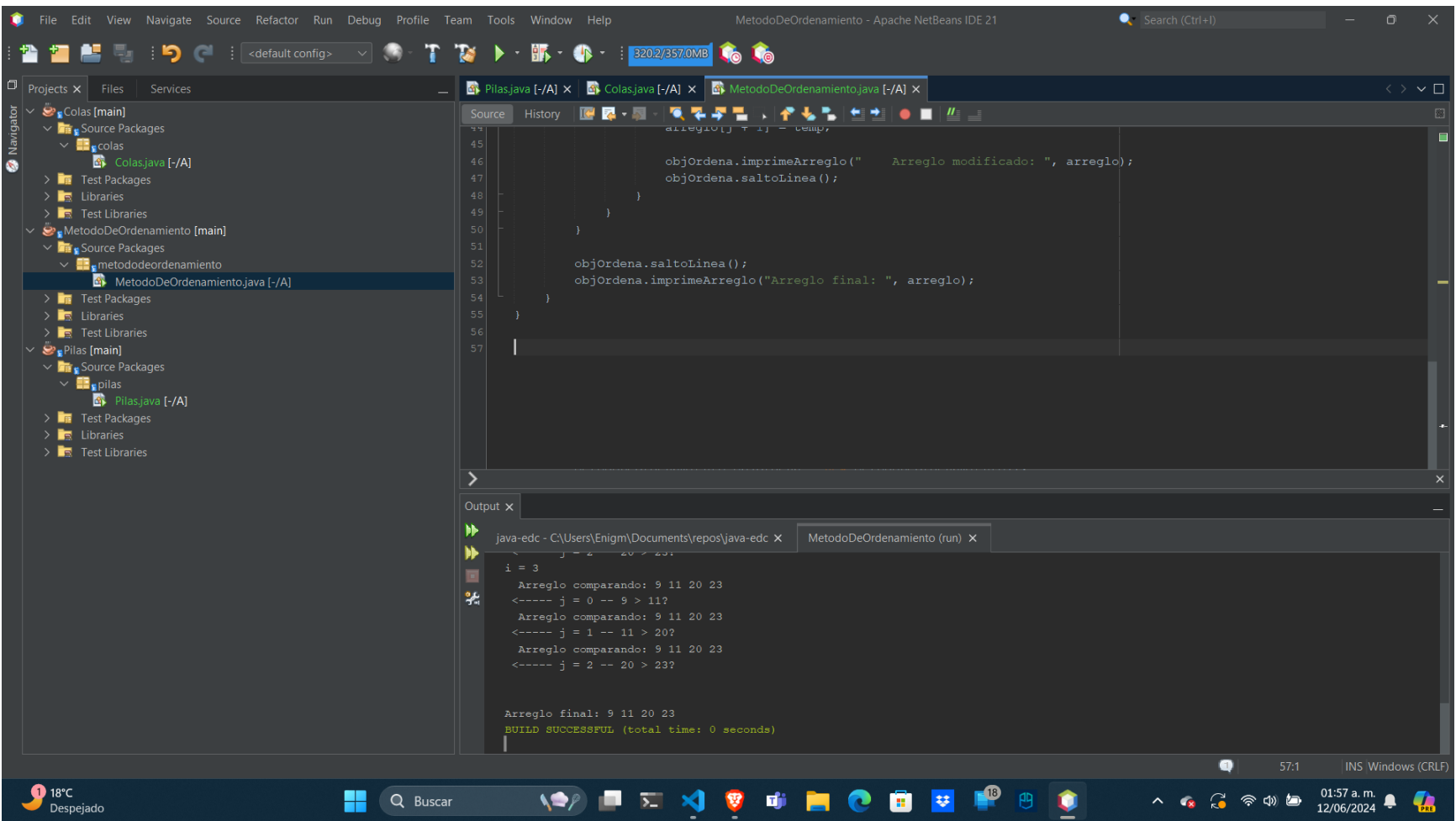


Figure 1: Captura del programa MetodoDeOrdenamiento.java