



Universidad Nacional Autónoma de México



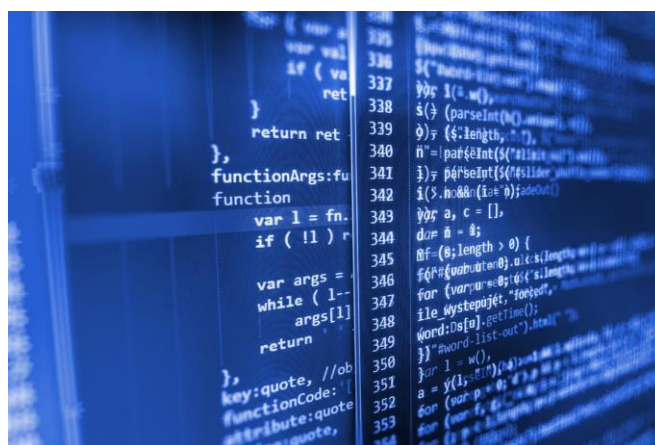
Facultad de Estudios Superiores Aragón

Ingeniería en Computación

Estructura de Datos

Jesús Hernández Cabrera

Axel Yahir Moreno Rodríguez



Turno Vespertino

Grupo 1360

```
J PotenciaRecursiva.java J PilaMedia.java J MergeSort.java
Estructura de Datos > Tareas > Tarea 12 > tarea12 > src > main > java > unam > mx > merge > J MergeSort.java > MergeSort > mezclar(int[], int[], int[])

1 package unam.mx.merge;
2 import java.util.Arrays;
3
4 public class MergeSort {
5     public static void main(String[] args) {
6         System.out.println(x:"----- MERGE -----");
7         int[] datos = {38, 27, 43, 3, 9, 82, 18, 19, 50, 61};
8         ordenarPorMezcla(datos);
9         System.out.println("Arreglo ordenado: " + Arrays.toString(datos));
10    }
11
12    public static void ordenarPorMezcla(int[] datos) {
13        if (datos.length > 1) {
14            int mitad = datos.length / 2;
15            int[] izquierda = Arrays.copyOfRange(datos, from:0, mitad);
16            int[] derecha = Arrays.copyOfRange(datos, mitad, datos.length);
17
18            System.out.println(Arrays.toString(izquierda) + " --- " + Arrays.toString(derecha));
19
20            ordenarPorMezcla(izquierda);
21            ordenarPorMezcla(derecha);
22
23            mezclar(datos, izquierda, derecha);
24        }
25
26        System.out.println("Regresando de recursión: " + Arrays.toString(datos));
27    }
28
29    private static void mezclar(int[] datos, int[] izquierda, int[] derecha) {
30        int i = 0, d = 0, k = 0;
31
32        while (i < izquierda.length && d < derecha.length) {
33            if (izquierda[i] < derecha[d]) {
34                datos[k++] = izquierda[i++];
35            } else {
36                datos[k++] = derecha[d++];
37            }
38        }
39
40        while (i < izquierda.length) {
41            datos[k++] = izquierda[i++];
42        }
43
44        while (d < derecha.length) {
45            datos[k++] = derecha[d++];
46        }
47    }
48
49 }
```



```
    }

    System.out.println("Regresando de recursión: " +
Arrays.toString(datos));
}

private static void mezclar(int[] datos, int[] izquierda, int[]
derecha) {
    int i = 0, d = 0, k = 0;

    while (i < izquierda.length && d < derecha.length) {
        if (izquierda[i] < derecha[d]) {
            datos[k++] = izquierda[i++];
        } else {
            datos[k++] = derecha[d++];
        }
    }

    while (i < izquierda.length) {
        datos[k++] = izquierda[i++];
    }

    while (d < derecha.length) {
        datos[k++] = derecha[d++];
    }
}
}
```