

# JAVA Programación Orientada a Objetos

## Algoritmo de Ordenamiento Burbuja

El algoritmo de ordenamiento burbuja es uno de los métodos más simples para organizar elementos en un arreglo. Este algoritmo puede aplicarse tanto a arreglos estáticos como a estructuras dinámicas similares, como colecciones del tipo `ArrayList`. Sin embargo, este texto se centrará exclusivamente en su implementación con arreglos estáticos para facilitar su comprensión.

---

### Funcionamiento del Algoritmo

#### 1. Comparación de Elementos:

- El algoritmo comienza comparando el primer elemento del arreglo con el siguiente.
- Si el primer elemento es mayor (o menor, dependiendo del orden deseado) que el segundo, se intercambian.
- Este proceso se repite para cada par de elementos adyacentes a lo largo del arreglo.

#### 2. Recorrido del Arreglo:

- Al finalizar un recorrido completo, el elemento más grande (o más pequeño) se desplazará hacia la última posición del arreglo.
- En la siguiente iteración, el proceso se aplica a los primeros  $n-1$  elementos, luego a los primeros  $n-2$ , y así sucesivamente, hasta que el arreglo esté ordenado.

#### 3. Detección de Optimización:

Una versión optimizada del algoritmo puede detectar si el arreglo está ordenado antes de completar todas las iteraciones, reduciendo el número de comparaciones.

# Ventajas y desventajas

## Ventajas

- Es sencillo de entender e implementar.
- Adecuado para arreglos pequeños o casi ordenados.
- Puede ordenar tanto datos numéricos como alfabéticos.

## Desventajas

- Tiene una complejidad temporal de  $O(n^2)$ , lo que lo hace ineficiente para arreglos grandes.
- No es adecuado para colecciones grandes o ya ordenadas.

## Ejemplos de Implementación

Ordenamiento de Números de Mayor a Menor:

```
public class BurbujaNumeros {
    public static void main(String[] args) {
        int[] numeros = {5, 3, 8, 6, 2};
        bubbleSortDesc(numeros);
        for (int num : numeros) {
            System.out.print(num + " ");
        }
    }
    public static void bubbleSortDesc(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] < arr[j + 1]) { // Cambio de menor
                    a mayor por mayor a menor
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}
```

## Ordenamiento de Strings de Menor a Mayor:

```
import java.util.Arrays;

public class BurbujaStrings{
    public static void main(String[] args) {
        String[] palabras = {"zorro", "abeja", "gato", "elefante", "perro"};
        bubbleSortAsc(palabras);
        System.out.println(Arrays.toString(palabras));
    }

    public static void bubbleSortAsc(String[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j].compareTo(arr[j + 1]) > 0) { // Comparación alfabética
                    String temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}
```

¡Anímate a implementar los ejemplos en tu editor de código y observa cómo funcionan en la práctica!