

Anexo 4: Ordenación en Java

Contenido

1.	Métodos de Ordenación	2
1.1.	Inserción	2
1.2.	Burbuja	3
1.1.1.	Burbuja	3
1.1.2.	Burbuja (método mejorado)	3
1.1.3.	Burbuja Optimizado	4
1.3.	Selección	5

1. Métodos de Ordenación

1.1. Inserción

```
public static void insercion(int[] a) {  
    int i, j, aux;  
  
    //Recorrido del array  
    for (i = 1; i < a.length; i++) {  
        aux = a[i];  
        for (j = i - 1; j >= 0 && aux < a[j]; j--) {  
            a[j + 1] = a[j];  
        }  
        a[j + 1] = aux;  
    }  
}
```

<https://liveexample.pearsoncmg.com/dsanimation/InsertionSortWithJava.html>

1.2. Burbuja

1.1.1. Burbuja

```
public static void burbuja(int[] a) {  
    int i, j, aux;  
    //Nº de vueltas  
    for (i = 0; i < a.length - 1; i++) {  
        //Recorrido del array  
        for (j = 0; j < a.length - 1 - i; j++) {  
            //En cada iteración el recorrido acaba un elemento antes  
            if (a[j] > a[j + 1]) {  
                aux = a[j];  
                a[j] = a[j + 1];  
                a[j + 1] = aux;  
            }  
        }  
    }  
}
```

1.1.2. Burbuja (método mejorado)

```
public static void burbuja(int[] a) {  
    int i, j, aux;  
    //Nº de vueltas  
    for (i = 0; i < a.length - 1; i++) {  
        //Recorrido del array  
        for (j = 0; j < a.length - 1 - i; j++) {  
            //En cada iteración el recorrido acaba un elemento antes  
            if (a[j] > a[j + 1]) {  
                aux = a[j];  
                a[j] = a[j + 1];  
                a[j + 1] = aux;  
            }  
        }  
    }  
}
```

1.1.3. Burbuja Optimizado

```
public static void burbujaM(int[] a) {  
    int aux, i = 0, j;  
    boolean intercambio = true;  
    // Mientras se realicen intercambios  
    while (intercambio) {  
        intercambio = false;  
        // Recorrido del array  
        for (j = 0; j < a.length - 1 - i; j++) {  
            // En cada vuelta el recorrido acaba un elemento antes  
            if (a[j] > a[j + 1]) {  
                aux = a[j];  
                a[j] = a[j + 1];  
                a[j + 1] = aux;  
                intercambio = true;  
            }  
        }  
        i++;  
    }  
}
```

<https://liveexample.pearsoncmg.com/dsanimation/BubbleSortWithJava.html>

1.3. Selección

```
public static void seleccion(int[] a) {  
    int i, j, aux, minimo;  
    //Recorrido del array  
    for (i = 0; i < a.length - 1; i++) {  
        minimo = i;  
        for (j = i + 1; j < a.length ; j++) {  
            if (a[j] < a[minimo]) {  
                minimo = j;  
            }  
        }  
        aux = a[i];  
        a[i] = a[minimo];  
        a[minimo] = aux;  
    }  
}
```

<https://liveexample.pearsoncmg.com/dsanimation/SelectionSortNeweBook.html>