

# Análisis del Problema 01

**A)** Este algoritmo se ejecutará hasta-desde veces.

```
private static int indiceMin(int[] A, int desde, int hasta) {  
    int indice = hasta - 1;  
  
    for (int i = desde; i < hasta; i++) {  
        if (A[i] < A[indice]) indice = i;  
    }  
  
    return indice;  
}  
  
/*  
 * Ordena el array A[0...A.length-1] de menor a mayor valor, usando el método  
 * indiceMin.  
 */  
private static void OrdenPorSeleccion(int[] A) {  
    for (int i = 0; i < A.length - 1; i++) {  
        int indiceMenor = indiceMin(A, i, A.length);  
        swap(A, i, indiceMenor);  
    }  
}
```

**B)** Este algoritmo hace uso del algoritmo anteriormente analizado.

Cada vez que entremos al bucle, llamaremos a la función `indiceMin`, la cual se ejecuta hasta-desde veces.

También usa la función `swap()`, pero esta tiene un coste de tiempo constante. El coste no incrementa con la entrada.

Por lo tanto el análisis del algoritmo de ordenación es el siguiente:

$$\left[ \begin{array}{cc} i = 0 & n - 0 \\ i = 1 & n - 1 \\ \vdots & \vdots \\ i = n - 1 & 1 \end{array} \right]$$

→ Se ejecuta n veces.

Progresión aritmética:

$$a1 + a2 + \dots + aN = \frac{(a1 + aN)n}{2}$$

Por lo tanto aplicando esta fórmula:

$$\frac{(n + 1)(n - 1)}{2} = \frac{n^2 - 1}{2}$$

Llegamos a la conclusión de que la complejidad es de:

$$O(n^2)$$