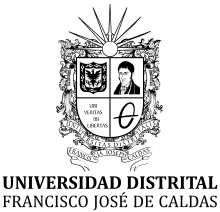
**COMPARACIÓN ENTRE LENGUAJES COMPILADORES E INTERPRETADORES**

****

DAVIDSON ESFLEIDER SANCHEZ GORDILLO - 20231020183

DIEGO FELIPE DIAZ ROA - 20201020147

DANIA LIZETH GUZMÁN TRIVIOÑO - 20221020061

JOSHOA ALARCON SANCHEZ - 20221020013

SEPTIEMBRE 2025

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

FACULTAD DE INGENIERÍA

CIENCIAS DE LA COMPUTACIÓN I

SIMAR ENRIQUE HERRERA JIMENEZ

**BUBBLE SORT**

El algoritmo de ordenamiento de burbuja o Bubble Sort consiste en comparar los elementos adyacentes en una lista de manera repetida, de tal forma de que si están en el orden incorrecto los intercambia y pasa al siguiente elemento de la lista, hasta que toda la lista esté ordenada

Este algoritmo posee un orden de complejidad cuadrático O(n^2) en el peor y el promedio de los casos, aunque es de complejidad O(n) en el mejor de los casos ya que solo deberá recorrer la lista una sola vez para corroborar que no más recorridos son necesarios.

**SELECTION SORT**

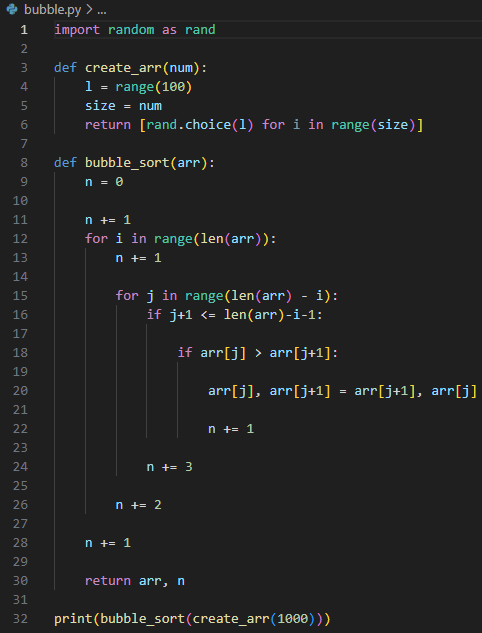
El algoritmo de ordenamiento por selección o Selection Sort consiste en tomar dos listas, una en la cual se encuentran los elementos no-sorteados y una en la que se colocarán los elementos sorteados. Se comprueba cada elemento de la lista para saber si es el mayor de la lista de elementos no-sorteados y lo intercambia con el primer elemento de la lista de los sorteados.

Este algoritmo posee un orden de complejidad cuadrático O(n^2) para todos los casos, incluyendo el peor, mejor y promedio, ya que siempre hará la comprobación para todos los elementos de la lista al menos una vez.

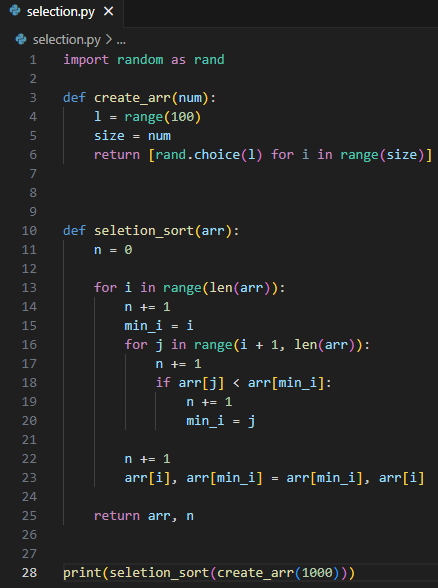
**ANÁLISIS COMPARATIVO**

Para la práctica de esta comparativa se usó el lenguaje Python y se escribieron los siguientes códigos:

* **Bubble Sort**

****

* **Selection Sort**

****

Ambos códigos tienen la capacidad de imprimir el número de pasos realizado durante la ejecución del programa con un array con elementos aleatorios en un rango introducido por el usuario, por lo cual se hicieron 5 ejecuciones en cada programa y se obtuvieron los siguientes resultados para el número de pasos:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **# de Ejecución** | **1** | **2** | **3** | **4** | **5** |
| **BUBBLE SORT** | 1760515 | 1748670 | 1759943 | 1751995 | 1753349 |
| **SELECTION SORT** | 505422 | 505561 | 505454 | 505524 | 505333 |

Como se puede observar en los resultados de los test, se puede ver que a pesar de que ambos algoritmos de ordenamiento tienen el mismo nivel de complejidad, a la hora de ejecutarlos se nota una clara diferencia de eficiencia entre los dos, siendo el algoritmo Selection Sort el más destacado.

Esto se debe debido a la diferencia entre los procesos de ordenamiento principalmente, ya que la cantidad de intercambios que se deben hacer en las listas es muy diferente entre los dos algoritmos, siendo que el algoritmo Selection Sort específicamente realiza n-1 intercambios para un array de n-elementos, mientras que en el algoritmo Bubble Sort el número de intercambios se ve incrementado debido a que los elementos se mueven de una posición a la vez mientras se hacen las comparaciones con los elementos adyacentes.