

## Hoja de Trabajo 6

### Colección del JCF:

Para la colección de los Pokémon del usuario, hemos elegido ArrayList. La razón para elegir ArrayList es que proporciona acceso rápido a los elementos por índice y es eficiente para iterar y ordenar. Dado que las operaciones principales que realizaremos en esta colección son agregar elementos, verificar si un elemento ya está presente y ordenar los elementos, ArrayList es una opción adecuada.

### Complejidad del Tiempo:

La operación #4 consiste en mostrar el nombre y tipo1 de todos los Pokémon existentes, ordenados por type1. La complejidad de tiempo depende de la implementación del Map que se esté utilizando, por ejemplo, el TreeMap:

TreeMap: Los valores en un TreeMap ya están ordenados por la clave. Sin embargo, como estamos ordenando por type1 y no por la clave, necesitamos recopilar los valores y ordenarlos, lo que tiene una complejidad de  $O(n \log n)$ .

**Link del repositorio:** <https://github.com/Luis-Angel-G/Hoja-de-Trabajo-6-Algoritmos-y-Estructura-de-Datos>

### Link del video:

[https://www.canva.com/design/DAGhq\\_f3mvk/s842qm2eJO9sj5li8BYWxg/edit?utm\\_content=DAGhq\\_f3mvk&utm\\_campaign=designshare&utm\\_medium=link2&utm\\_source=sharebutton](https://www.canva.com/design/DAGhq_f3mvk/s842qm2eJO9sj5li8BYWxg/edit?utm_content=DAGhq_f3mvk&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)