**PRACTICA 5 |ARREGLOS**

**Introducción**

Para esta práctica se seguirá creando código para manipulación de arreglos de forma conveniente. En el caso de este ejercicio particular la clase a crear ya no solamente tendrá el arreglo de números en el orden de cómo se fueron insertando, sino que ahora también los valores tendrán que estar ordenados en todo momento y no permitiendo elementos duplicados. Entonces pues, a través de la implementación que daré a las clases y sus respectivos métodos requeridos en la práctica pretendo mejorar mis habilidades de manipulación, razonamiento y comprensión de los arreglos.

**Desarrollo**

En esta práctica nuevamente se tuvieron que crear algunas clases y paquetes, así como ha ido pasando en algunas de las anteriores. Para este ejercicio el principal foco de atención fue una clase nueva llamada **ArregloOrdenado**, que, como podría esperarse, hereda de **Arreglo** y que tiene la facultad, además de asegurar el orden de los valores contenidos en su interior, de evitar que haya datos duplicados.

Con ayuda del profesor, en clase se realizó el análisis y pseudocódigo de algunos de los métodos requeridos en la práctica, a los cuales se les hizo una pequeña modificación para que los datos dentro del arreglo estuvieran ordenados de forma ascendente o descendente según el parámetro indicado en el constructor de la clase **ArregloOrdenado.** Las funciones restantes fueron más cortas y sencillas de programar. A continuación, se indican los archivos modificados o creados.

**ArregloOrdenado.java (paquete edlineal):**

* public boolean agregar(Object valor).
* public Object buscar(Object valor).
* public boolean setElemento(Object valorV, Object valorN, int numOcurrencias).
* public boolean setElemento(int pos, Object valor).
* public Object eliminar(Object valor).
* public boolean agregarArreglo(Arreglo arreglo2).
* public void invertir().
* public void rellenar(Object valor, int cuantos).

Cabe mencionar que todos los métodos utilizados han sido sobrescritos en esta clase; algunos provienen de la interfaz **Lista** y otros son propios de la clase **Arreglo**.

**CriterioOrden.java (paquete utilidades):**

* public String getNombre().
* public int getId().

En este enum se definieron los atributos ascendente y descendente para el parámetro de la clase **ArregloOrdenado**.

**Comparador.java (paquete utilidades).**

* public static int comparar(Object obj1, Object obj2).
* public static int comparer(Number numero1, Number numero2).

Esta clase fue sumamente necesaria para realizar las comparaciones de si un objeto es más grande o pequeño que otro dentro de la clase **ArregloOrdenado.**

**Arreglo.java (paquete edlineal).**

* public boolean insertar(int pos, Object valor).
* public boolean copiarArreglo(Arreglo arreglo2).

**PruebaArregloOrdenado.java (paquete interfaz).**

La clase anterior es donde se encuentran las pruebas de esta práctica.

**Para mayor información sobre los métodos y la clase creada vaya a la ruta: /docs/index.html dentro del proyecto.**

**Conclusiones**

Al inicio de la práctica cuando aún estábamos en clase aprendí algo nuevo y es de la existencia de clases de tipo “enum”, que no recuerdo haber ni siquiera oído mencionar anteriormente.

Hablando un poco más acerca del funcionamiento principal, me topé con ciertas dificultades principalmente a la hora de hacer uso de mi clase de utilidad **Comparador** lo que me retrasó bastante porque era sumamente fundamental para el funcionamiento de otros métodos. Después de un tiempo logré cubrir el error y seguir codificando los demás métodos donde en algunos seguí teniendo dificultad para comprender algún proceso o por errores.

Puedo decir que aún me falta mejorar en cuanto a la manipulación de arreglos y espero seguir perfeccionando mis habilidades conforme pasan las prácticas y el transcurso de la materia.