

## **Carrera de algoritmos.**

### **Laboratorio final de Algoritmos y programación 2.**

#### **Requerimientos funcionales.**

**Requerimiento número 1.** Agregar datos de manera iterativa; permite agregar una cantidad aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 2.** Agregar datos de manera recursiva; permite agregar una cantidad de aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 3.** Buscar datos de manera iterativa; permite buscar una cantidad aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 4.** Buscar datos de manera recursiva; permite buscar una cantidad aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 5.** Eliminar datos de manera iterativa; permite eliminar una cantidad aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 6.** Eliminar datos de manera recursiva; permite eliminar una cantidad aleatoria de datos especificada en la interfaz de usuario en cada una de las estructuras de datos del programa.

**Requerimiento número 7.** Mostrar el tiempo en cronómetros por cada algoritmo para agregar datos en las estructuras de datos. En la interfaz de usuario se mostrarán 3 cronómetros que medirán el tiempo que tarda cada algoritmo para agregar los datos aleatorios en las estructuras de datos.

**Requerimiento número 8.** Mostrar el tiempo en cronómetros por cada algoritmo para buscar datos en las estructuras de datos. En la interfaz de usuario se mostrarán 3 cronómetros que medirán el tiempo que tarda cada algoritmo para buscar los datos aleatorios en las estructuras de datos.

**Requerimiento número 9.** Mostrar el tiempo en cronómetros por cada algoritmo para eliminar datos en las estructuras de datos. En la interfaz de usuario se mostrarán 3 cronómetros que medirán el tiempo que tarda cada algoritmo para eliminar los datos aleatorios en las estructuras de datos.

#### **Requerimientos no funcionales.**

**Requerimiento número 1.** Los elementos que van a ser almacenados, buscados y eliminados en las estructuras de datos "ArrayList", "Linked List" y "Binary search

tree” son de tipo “long”. Además, se debe tener en cuenta los valores máximos y mínimos de este tipo de dato.

**Requerimiento número 2.** La interfaz de usuario del programa deberá ser desarrollada exclusivamente con la librería de JavaFX.

**Requerimiento número 3.** Los métodos que se encargan de buscar y eliminar en valores en las estructuras de datos retornaran valores de tipo “boolean”.

**Requerimiento número 4.** Cada estructura de datos de la competencia contara con su propio hilo para almacenar, buscar y eliminar valores de manera iterativa y recursiva.

**Requerimiento número 5.** El hilo del reloj que se muestra en la interfaz tendrá su propio hilo de ejecución.

**Requerimiento número 6.** Los dos círculos que cambian su tamaño de manera sincronizada en la interfaz de usuario se ejecutaran en un solo hilo de ejecución.

**Requerimiento número 7.** El reloj que esta en la interfaz de usuario se ejecutara en un hilo de ejecución.

