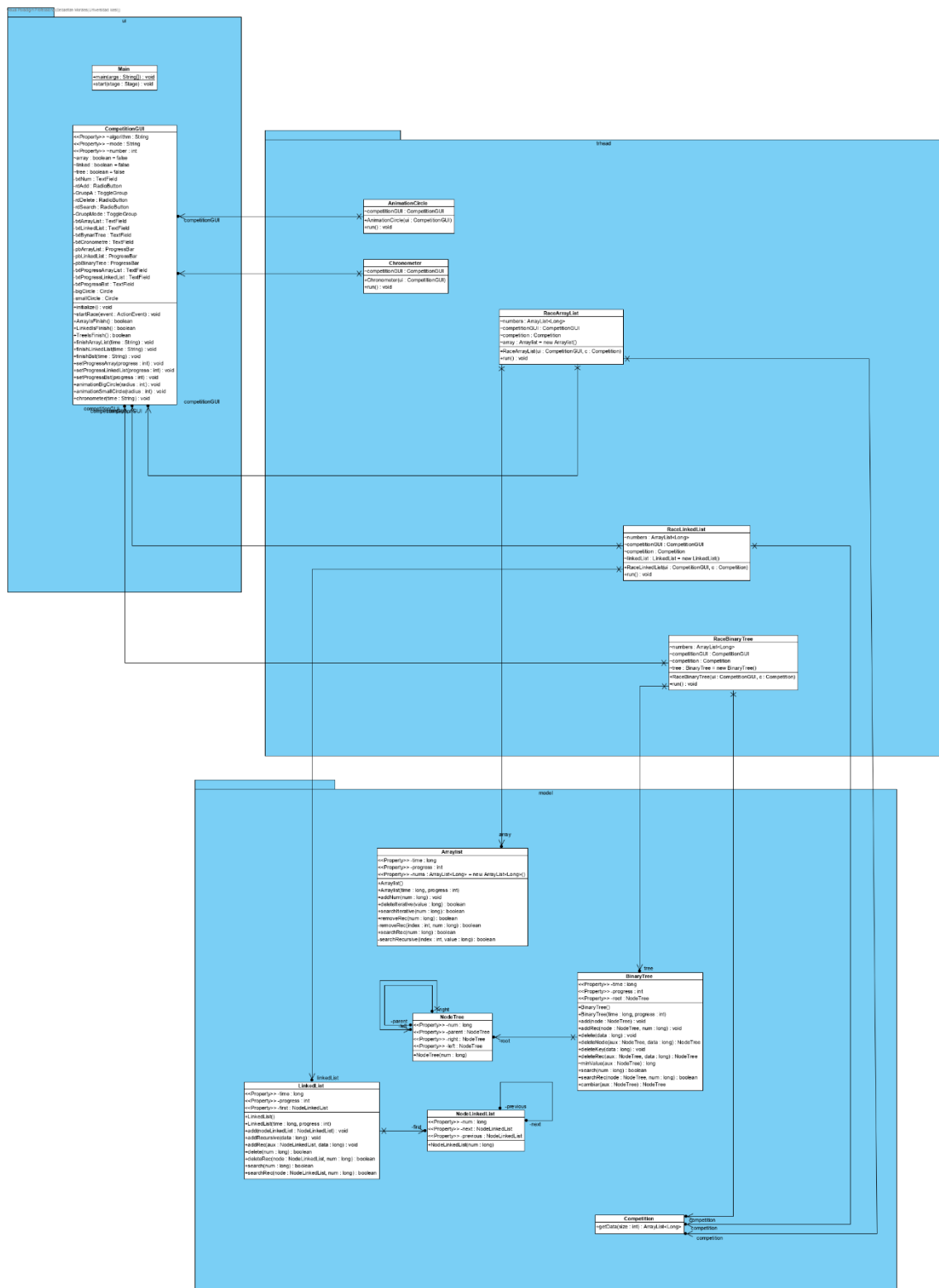


## Requerimientos funcionales



El sistema debe permitir:

- Ingresar el usuario una N cantidad de números, para que el programa a partir de este valor genere esa cantidad N de valores aleatorios. Devolviendo así los N elementos para los diferentes algoritmos.
- La carrera debe permitir dos modos de competencia tanto iterativo como recursivo.
- Agregar N cantidad de elementos a cada estructura de datos (Arraylist, Lista Enlazada, Árbol Binario de Búsqueda) se debe tener un método de agregar el cual consiste en añadir N elementos generados aleatoriamente a la estructura de datos, cuando se agregue la cantidad de elementos.
- Buscar N cantidad de elementos en cada estructura de datos (Arraylist, Lista Enlazada, Árbol Binario de Búsqueda) estas deben tener un método de buscar el cual permite consultar N elementos generados aleatoriamente sobre una estructura de datos previamente creada con N elementos también aleatorios. Los resultados de cada búsqueda retornaran un dato booleano que, si el elemento existe o que no existe.
- Eliminar N cantidad de elementos en cada estructura de datos (Arraylist, Lista Enlazada, Árbol Binario de Búsqueda) estas deben tener un algoritmo el cual consiste en eliminar N elementos generados aleatoriamente sobre una estructura de datos previamente creada con N elementos aleatorios. Este algoritmo me debe retornar un dato booleano, que si el elemento se elimina o no se elimina.
- Cada algoritmo de agregar, eliminar y buscar al finalizar su ejecución debe mostrar su tiempo de ejecución. Puesto que este definirá el ganador de la carrera.
- La carrera debe contar con un cronometro, el cual llevara el tiempo con el que inicio la carrera hasta donde termina.
- Mostrar 3 barras progresivas, donde cada una corresponde a la estructura de datos, estas deben ir llenándose progresivamente, mostrando el avance de la carrera.

### Requerimientos no funcionales:

- Los métodos de agregar, buscar, y eliminar deben ser iterativos, para cada estructura de datos.
- los métodos de agregar, buscar, y eliminar también deben ser recursivos, para cada estructura de datos.
- Debe contar con una animación de un círculo que se reduce poco hasta un punto y luego crece hasta el tamaño original del círculo y luego vuelve a reducirse. Además de que hay otro círculo que hará lo mismo, pero, al contrario, reduciéndose cuando el otro aumenta y viceversa, por último esta animación funcionara hasta que termine la carrera.
- La interfaz del usuario tiene que ser de fácil manejo para el usuario.
- El tiempo se debe reiniciar apenas acabe la carrera, para volver a ingresar, a otra carrera, cada competencia es un hilo.