**Requerimientos Funcionales**

El programa debe estar en la capacidad de:

1. Permitir la competencia entre estructuras de datos
   1. Las estructuras de datos son:
      1. ArrayList con elementos tipo Long
      2. Lista enlazada con elementos tipo Long
      3. Arbol binario de busqueda con elementos tipo Long
   2. El ganador será la estructura que en el menor tiempo haga las siguientes operaciones
      1. Agregar agregar N elementos generados aleatoriamente a la estructura de datos.
         1. Agregar-Iterativo
         2. Agregar-Recursivo
         3. En la estructura de datos de ArrayList no aplica agregar iterativo ni agregar recursive.
      2. Consultar consultar N elementos generados aleatoriamente sobre una estructura de datos previamente creada con N elementos también aleatorios. Solo se toma el tiempo de las N consultas.Los resultados de cada consulta retornarán que el elemento existe o que no existe (boolean), pero eso no será tenido en cuenta en la carrera.
         1. Consultar-Iterativo
         2. Consultar-Recursivo
      3. Eliminar N elementos generados aleatoriamente sobre una estructura de datos previamente creada con N elementos aleatorios. Solo se toma el tiempo de las N eliminaciones. Los resultados de la eliminación podrá ser que el elemento se elimina o que no se elimina porque no existe, pero ese resultado no será tenido en cuenta en la carrera.
         1. Eliminar-Iterativo
         2. Eliminar-Recursivo
      4. Donde I significa Iterativo y R significa Recursivo, las posibles competencias son:
         1. AL - A – I vs LE - A – I vs ABB - A – I
         2. AL - C – I vs LE - C – I vs ABB - C – I
         3. AL - E – I vs LE - E – I vs ABB - E – I
         4. AL - A – R vs LE - A – R vs ABB - A – R
         5. AL - C – R vs LE - C – R vs ABB - C – R
         6. AL - E – R vs LE - E – R vs ABB - E - R
2. Se deberá mostrar una animación durante la Carrera
   1. En el momento en que inicia la carrera, también iniciará un cronómetro que contará el tiempo en minutos:segundos:milisegundos transcurridos desde que inició la carrera. También iniciará una pequeña animación de un círculo que se reduce poco a poco hasta un punto y luego crece hasta el tamaño original del círculo y luego vuelve a reducirse, así hasta que termine la carrera y el último competidor termine.
   2. OItro círculo en la misma posición que hará lo mismo pero al contrario, se reduce cuando el otro aumenta y al contrario.
3. Se deberá llevar el cronómetro en minutos:segundos:milisegundos transcurridos desde que inició la carrera
   1. Para toda la carrera
   2. Para cada una de las estructuras de datos
4. El usuario debe estar en la capacidad de:
   1. Digitar el tamaño de la carrera o el N número de elementos presentes en las estructuras de datos
   2. Escoger entre los tres tipos de algoritmos
      1. Agregar
      2. Consultar
      3. Eliminar
   3. Escoger entre los dos tipos de modo
      1. Iterativo
      2. Recursivo
   4. Una vez digitado toda la información, pueda comenzar la carrera al presionar el botón que diga Run!
5. Para cada una de las estructuras de datos, debe de tener una imagen que lo represente.
6. La ventana en la cual se presente la carrera no se deberá en ningún momento bloquear durante la carrera.