

INFOME DEL LABORATORIO

REQUERIMIENTOS FUNCIONALES

- RF1.** Leer la cantidad de datos que se pueden agregar para la ejecución del programa. Dando la cantidad numérica de cuantos datos se desean guardar.
- RF2.** Seleccionar que modalidad se desea correr. Mostrando las opciones de competición elegir la disciplina y la modalidad para realizar la competición.
- RF3.** Mostrar un cronometro que indique cuanto tiempo ha transcurrido de la prueba. Que incluye una figura circular que gira mientras el cronometro corre. El cronometro mostrara el tiempo en minutos : segundos : milisegundos.
- RF4.** Mostrar el resultado final de cuanto duro cada “deportista” en realizar su prueba. En donde se dirá el nombre del deportista y su tiempo en minutos : segundos : milisegundos.
- RF5.** Mostrar la bandera de cada deportista encima de sus resultados.

REQUERIMIENTOS NO FUNCIONALES

1. Agregar datos aleatorios de una cantidad dada por el usuario en los siguientes tipos de base de datos.
 - 1.1. Agregar en un ArrayList de manera ordenada, de modo que el primero que aparezca sea el menor y el ultimo dato el mayor. Sí los datos se repiten se agregará justo delante del ultimo dato que cumpla la condición de ser menor al que se va a agregar, y todos los demás se correrán 1 posición.
 - 1.2. Agregar en una lista doblemente enlazada no circular en el orden que vaya saliendo, es decir, que la lista no estará ordenada.
 - 1.3. Agregar en un árbol binario de búsqueda de modo que el primer dato agregado sea la raíz de dicho árbol y sobre el se vayan agregando a la derecha si es mayor al primer dato o a la izquierda si es menor al dato. En caso de que se agrega un dato igual esta ira a la izquierda.
2. Buscar datos aleatorios de una cantidad dada por el usuario, donde en cuanto se encuentre el dato se terminará de buscar ese dato, en los siguientes tipos de base de datos
 - 2.1. Buscar datos previamente creados en un ArrayList con el método de búsqueda binaria, debido a que los datos son agregados de manera ordenada. Esto quiere decir, que se buscara por la mitad del arreglo y si es menor al que se busca se coge la mitad derecha del arreglo, si es mayor se busca en la mitad izquierda del arreglo y así sucesivamente hasta encontrar el dato, o llegar a solo 1 espacio del arreglo y se verifique si es o no.

- 2.2. Buscar datos previamente creados en una lista doblemente enlazada con el método de búsqueda secuencial, lo que implica que busca desde el primer dato hasta el ultimo dato en la secuencia.
 - 2.3. Buscar datos previamente creados en un árbol binario de búsqueda. La búsqueda será de tipo binario acorde con el tipo de base de datos en la cual que realiza la búsqueda.
3. Eliminar la cantidad de elementos aleatorios dado por el usuario en cada uno de los siguientes tipos de datos
 - 3.1. Eliminar datos previamente creados en un ArrayList, buscando el dato a eliminar y de encontrarlo, eliminara la posición en la que se encuentra
 - 3.2. Eliminar datos previamente creados en una lista doblemente enlazada no circular, buscando el dato a eliminar y de encontrarlo, se le asignara al anterior como siguiente nodo el siguiente del dato a eliminar.
 - 3.3. Eliminar datos previamente creados en un árbol binario de búsqueda, buscando al dato que se va a eliminar
4. Adicionalmente, se realizarán estas acciones de 2 formas diferentes, de forma iterativa y de forma recursiva, dependiendo de lo que elija el usuario se usara una u otra.

DIAGRAMA DE CLASES

