

REQUERIMIENTOS FUNCIONALES:

R1: Agregar datos aleatoriamente a un ArrayList. Se agrega una cantidad N de datos a un ArrayList, estos datos se generarán aleatoriamente.

R2: Agregar datos aleatoriamente a una lista enlazada de manera iterativa. Se agrega una cantidad N de datos a una lista enlazada, estos datos se generarán aleatoriamente.

R3: Agregar datos aleatoriamente a un árbol binario de búsqueda de manera iterativa. Se agrega una cantidad N de datos a un árbol binario de búsqueda, estos datos se generarán aleatoriamente.

R4: Agregar datos aleatoriamente a una lista enlazada de manera recursiva. Se agrega una cantidad N de datos a una lista enlazada, estos datos se generarán aleatoriamente.

R5: Agregar datos aleatoriamente a un árbol binario de búsqueda de manera recursiva. Se agrega una cantidad N de datos a un árbol binario de búsqueda, estos datos se generarán aleatoriamente.

R6: Eliminar datos de ArrayList de manera recursiva. Se elimina una cantidad N de datos a un ArrayList, tanto los datos ya ingresados al arrayList como los que se van a eliminar ya han sido generados anteriormente para que estos procesos no interrumpen en la competición.

R7: Eliminar datos de una lista enlazada de manera recursiva. Se elimina una cantidad N de datos de una lista enlazada, tanto los datos ya ingresados a la lista enlazada como los que se van a eliminar ya han sido generados anteriormente para que estos procesos no interrumpen en la competición.

R8: Eliminar datos a un árbol binario de búsqueda de manera recursiva. Se elimina una cantidad N de datos a un ABB, tanto los datos ya ingresados al ABB como los que se van a eliminar ya han sido generados anteriormente para que estos procesos no interrumpen en la competición.

R9: Eliminar datos de ArrayList de manera iterativa. Se elimina una cantidad N de datos a un ArrayList, tanto los datos ya ingresados al arrayList como los que se van a eliminar ya han sido generados anteriormente para que estos procesos no interrumpen en la competición.

R10: Eliminar datos de una lista enlazada de manera iterativa. Se elimina una cantidad N de datos de una lista enlazada, tanto los datos ya ingresados a la lista enlazada como los que se van a eliminar ya han sido generados anteriormente para que estos procesos no interrumpen en la competición.

R11: Eliminar datos a un árbol binario de búsqueda de manera iterativa. Se elimina una cantidad N de datos a un ABB, tanto los datos ya ingresados al ABB como los que se van a

eliminar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R12: Consultar datos a un ArrayList de manera iterativa. Se consulta una cantidad N de datos a un ArrayList, tanto los datos ya ingresados al ArrayList como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R13: Consultar datos a una lista enlazada de manera iterativa. Se consulta una cantidad N de datos a una lista enlazada, tanto los datos ya ingresados a la lista enlazada como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R14: Consultar datos a un árbol binario de búsqueda de manera iterativa. Se consulta una cantidad N de datos a un ABB, tanto los datos ya ingresados al ABB como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R15: Consultar datos a un ArrayList de manera recursiva. Se consulta una cantidad N de datos a un ArrayList, tanto los datos ya ingresados al ArrayList como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R16: Consultar datos a una lista enlazada de manera recursiva. Se consulta una cantidad N de datos a una lista enlazada, tanto los datos ya ingresados a la lista enlazada como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R17: Consultar datos a un árbol binario de búsqueda de manera recursiva. Se consulta una cantidad N de datos a un ABB, tanto los datos ya ingresados al ABB como los que se van a consultar ya han sido generados anteriormente para que estos procesos no interrumpan en la competición.

R18: Llevar el conteo del tiempo mediante la creación de un cronometro, el cual se iniciará cuando inicie la carrera y terminara cuando todos los miembros hallan llegado a la línea de meta (terminen de ejecutarse).

R19: Mostrar el tiempo que se demora cara uno de los algoritmos en terminarse.

R20: Reestablecer el programa, devuelve el programa a su

REQUERIMIENTOS NO FUNCIONALES:

-Los algoritmos se correrán paralelamente al mismo tiempo mediante el uso de hilo, por ejemplo si se escoge el de agregar recursivo entonces se ejecutarán el de agregar recursivo del ArrayList, de la lista enlazada y de el árbol binario de búsqueda.

-El cronometro se ejecutará como un hilo.

