Dennis Cheong 11241278

Oscar Nahin 11211323

**Rendimiento de las implementaciones del TDALista**

**Introducción**

En el área de programación existen terminos como el TDA (tipos de datos abstractos), que se puede pensar como un modelo matemático con una serie de operaciones definidas en ese modelo.(Aho,Hopcroft,& Ullman,2015). Uno de los TDA más fundamentales son las listas, el TDALista nos permite representar una colección de datos de forma organizada. Estos tienen 3 tipos de implementaciones y dentro de cada implementación tienen funciones que se irá discutiendo a medida vaya avanzando en la lectura.

El presente artículo de investigación explica y analiza el rendimiento que tiene los 3 tipos de implementaciones del TDALista. Según (definición abc, 2015) rendimiento es el producto o la utilidad que rinde o da una persona o cosa. Para esta investigación, rendimiento significará que tan rápido o cuantos recursos le tomará las funciones de las implementaciones para completarse.

A continuación se explorará a más profundidad estos nuevos e importantes términos de la programación.

**Marco teórico**

Como se había mencionado anteriormente, uno de los TDA más fundamentales es la lista. Una lista define una secuencia de elementos donde cada uno de ellos tienen una posición.(Aho,Hopcroft,& Ullman,2015). Estas listas poseen varias operaciones que se utilizaron para la investigación de este artículo, de las cuales las más importantes se describen a continuación:

\*insert(p): ingresa un elemento a la lista en la posición *p.*

\*remove(p): elimina un elemento de la lista en la posición *p.*

\*get(p): retorna el elemento ubicado en la posición *p* de la lista.

\*indexOf(E): retorna la posición del elemento *E* en la lista.

\*first(): retorna el primer elemento de la lista.

\*last(): retorna el último elemento de la lista.

También, se especificó la existencia de 3 tipos de implementación que son las *listas mediante arreglos* (pueden ser de tamaño fijo o variable), las *listas enlazadas* (pueden ser sencillas o doble) y las *listas mediante cursores* ( {tamaño fijo, tamaño variable} x {sencillas, doble} ).

Según (Aho,Hopcroft,& Ullman,2015) en la implementación de las listas, las listas mediante arreglos, tienen elementos que se almacenan en celdas contiguas de un arreglo. La segunda forma, listas enlazadas, utilizan apuntadores para enlazar elementos consecutivos. Y por último, las listas basadas en cursores, hacen uso de los cursores para simular los apuntadores para enlazar los elementos.

Cada implementación tiene su propia ventaja y desventaja. Las listas mediante arreglos permite el facil recorrido dentro de ella, pero al ingresar o eliminar un elemento dentro de ella requiere desplazar un posición dentro del arreglo. Las listas enlazadas evita este problema de desplazamiento y el empleo de memoria para almacenar una lista, pero requiere de un espacio adicional para los apuntadores. Las listas mediante cursores se utilizan generalmente para lenguajes de programación que no poseen apuntadores, usando este tipo de implementación permite la simulación de ellas.

**Metodología**

Para obtener resultados de manera más rápida se utilizó FSArrayList(lista mediante arreglo con tamaño fijo), DLLinkedList(lista doblemente enlazada) y FSCursorList(lista mediante cursores con tamaño fijo).

Con la ayuda del lenguaje de programación c++, se elaboró un programa que analizaba el rendimiento de cada implementación del TDALista. Se pudo lograr este análisis con la ayuda de las diferentes operaciones que poseen estas listas, que al sacar el resultado, se pudo comparar el rendimiento de cada función por cada implementación.

Para facilitar el análisis y observación del rendimiento de cada implementacion,se utilizaron varias gráficas. Para elaborar una gráfica es necesario recopilar datos, para ello, se calculó mil veces el tiempo en que se tardaba realizar cada función de los 3 tipos de implementaciones en posiciones aleatorias. Estos resultados se iban guardando en distintos archivos dependiendo de la implementación y la función. Para evitar tener gráficas con mucho dato, los resultados se iban almacenando cada vez que “i” en el ciclo “for” fuese un múltiplo de 10. Por ende, se recolectaron 100 valores para cada función en cada implementación. Una vez terminado estos procesos, se utilizó Microsoft Excel para la elaboración de las gráficas y con ello proceder a la conclusión.

**Resultado**

**Conclusión**

Con los resultados obtenido, las funciones de la lista enlazada poseen mejor rendimiento sobre las otras implementaciones. Esto puede ser debido a lo que se había mencionado anteriormente, que las listas enlazadas no requieren guardar un espacio de memoria para una lista, sino que puede obtener cualquier información directamente a través de los nodos.

Para los resultados de la lista mediante cursores, se puede observar que requiere de más tiempo para su ejecución. La simulación de los apuntadores mediante los cursores podría ser la causa para tales tiempo de ejecución. Se sabe que no todos los lenguajes de programación poseen apuntadores, si se puede evitar el uso de las listas mediante cursores, es mejor evitarlos.

En los resultados, el rendimiento de las listas mediante arreglo no fueron del todo desastroso. Aunque las listas mediante arreglo no son tan eficiente por el recorrido que tiene que realizar para buscar los elementos, este recorrido es lo que lo hace facil y comodo de utilizar.

Sin lugar a duda, cada implementación posee ventajas y desventajas. Para realizar un programa con buen rendimiento, según esta investigación, es mejor utilizar las listas enlazadas. Para realizar un programa de manera rapida, una buena opción serían las listas mediante arreglos. Y por último si es necesario, utilizen las listas mediante cursores.

**Bibliografía**

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2015). El tipo de datos abstracto lista. In *Estructuras de Datos y Algoritmos* (pp. 38-39). Mexico: Addison Wesley.

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2015). Realización de listas. In *Estructura de Datos y Algoritmos* (pp. 41-51). Mexico: Addison Wesley.

Aho, A. V., Hopcroft, J. E., & Ullman, J. D. (2015). Tipos de datos abstractos. In *Estructuras de Datos y Algoritmos* (pp. 10-11). Mexico: Addison Wesley.

*definición abc*. (2015). Retrieved from http://www.definicionabc.com/general/rendimiento.php