

## Resultados obtenidos UT3 TA13

### Inserción al principio

```
Duración TDA LISTA - inserta al principio: (ms)-157
terminada la insercion al principio en mi lista
Duración LINKEDLIST - inserta al principio: (ms)-169
terminada la insercion al principio en una linked list
Duración ARRAYLIST - inserta al principio: (ms)-21774
terminada la insercion al principio en una array list
```

Observamos que la TDA Lista desarrollada por nosotros y LinkedList proporcionada por Java tienen resultados similares. Explicamos esto porque ambos métodos son de orden constante, ya que los dos tienen una referencia al primero de la lista.

En el ArrayList para insertar al principio tenemos que mover todos los elementos, lo cual es mucho menos eficiente, siendo de orden lineal.

En estos casos sería mejor utilizar una lista enlazada simple, como la TDA Lista o LinkedList.

### Inserción al final

```
comienza la insercion al final de las tres listas
Duración TDA LISTA - inserta al final: (ms)-146864
terminada la insercion al final en mi lista
Duración LINKEDLIST - inserta al final: (ms)-123
terminada la insercion al final en una linked list
Duración ARRAYLIST - inserta al final: (ms)-127
terminada la insercion al final en una array list
```

En este caso nos encontramos con una diferencia importante entre TDA Lista y LinkedList.

La diferencia entre estos radica en que LinkedList tiene una referencia al último nodo, permitiendo una inserción directa al final (Orden Constante), mientras TDA Lista posee una referencia solo al primero nodo, requiriendo recorrer toda la lista para insertar al final (Orden Lineal).

En el caso del ArrayList, este también tiene una referencia al final permitiendo una inserción directa de Orden Constante.

Para la inserción al final concluimos que lo mejor es utilizar una TDA Lista con doble referencia (primer y último nodo) o utilizar LinkedList o ArrayList.

### Recorrido

```
comienza el recorrido de las listas
Duración recorrido TDA LISTA: (ms)-28
terminada de recorrer mi lista
Duración recorrido LINKEDLIST: (ms)-5880
terminada de recorrer la linked list
Duración recorrido ARRAYLIST: (ms)-13
terminada de recorrer la array list
```

Observamos en el recorrido que el método utilizado en TDA Lista fue más eficiente que el método utilizado para recorrer la LinkedList, ya que para recorrer esta última se necesita pasarle el índice del lugar donde se encuentra el nodo. Esto requiere realizar una búsqueda por cada iteración. Esta búsqueda es de Orden Lineal.

En cambio, para recorrer la TDA Lista se hace una operación de Orden Constante (se utiliza la referencia al nodo siguiente), no siendo necesario una búsqueda por cada elemento.

El ArrayList tiene referencia a todos los elementos de la lista, por lo tanto para devolverlo es una operación de Orden Constante. En este caso no es necesario realizar una búsqueda.

Concluimos que para recorrer las listas es más eficiente utilizar un TDA Lista o ArrayList.