



**UNIDAD PROFESIONAL INTERDISCIPLINARIA
DE INGENIERÍA CAMPUS ZACATECAS.**

INGENIERÍA EN SISTEMAS COMPUTACIONALES.

LIST, ARRAYLIST Y LINKEDLIST

ALUMNO: ARTURO DEL BOSQUE DÍAZ DE LEÓN.

2CM1

PROGRAMACIÓN ORIENTADA A OBJETOS

MAESTRO: ROBERTO OSWALDO CRUZ LEIJA

24 / Octubre / 2019

LIST

Esta interfaz también conocida como “secuencia” normalmente acepta elementos repetidos o duplicados, y al igual que los arrays es lo que se llama “basada en 0”. Esto quiere decir que el primer elemento no es el que está en la posición “1”, sino en la posición “0”.

Debido a la gran variedad y tipo de listas que puede haber con distintas características como permitir que contengan o no elementos null, o que tengan restricciones en los tipos de sus elementos, hay una gran cantidad de clases que implementan esta interfaz.

ARRAYLIST

La clase ArrayList en Java, es una clase que permite almacenar datos en memoria de forma similar a los Arrays, con la ventaja de que el número de elementos que almacena, lo hace de forma dinámica, es decir, que no es necesario declarar su tamaño como pasa con los Arrays.

LINKEDLIST

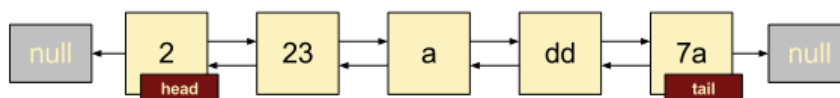
Ésta se basa en la implementación de listas doblemente enlazadas. Esto quiere decir que la estructura es un poco más compleja que la implementación con ArrayList, pero... ¿Qué beneficios nos aporta si la estructura es más compleja?

LinkedList y ArrayList son dos diferentes implementaciones de la interfaz List. LinkedList usa internamente una lista doblemente ligada, mientras que ArrayList usa un arreglo que cambia de tamaño dinámicamente.

Podemos visualizarlas de la siguiente forma:

Array vs. Linked List

Linked List



Array



LinkedList permite eliminar e insertar elementos en tiempo constante usando iteradores, pero el acceso es secuencial por lo que encontrar un elemento toma un tiempo proporcional al tamaño de la lista.

Por otro lado, ArrayList ofrece acceso en tiempo constante, pero si quieres añadir o remover un elemento en cualquier posición que no sea la última es necesario mover elementos. Además, si el arreglo ya está lleno es necesario crear uno nuevo con mayor capacidad y copiar los elementos existentes.

Referencias:

- <http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/>
- <https://jarroba.com/arraylist-en-java-ejemplos/>
- https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=603:interface-list-del-api-java-clases-arraylist-linkedlist-stack-vector-ejemplo-con-arraylist-cu00920c&catid=58&Itemid=180
- <https://es.stackoverflow.com/questions/172954/cuando-es-mejor-usar-linkedlist-y-cuando-arraylist>
- https://www.aprenderaprogramar.com/index.php?option=com_content&view=article&id=605:interface-list-clase-linkedlist-api-java-ejercicio-diferencias-entre-arraylist-y-linkedlist-codigo-cu00921c&catid=58&Itemid=180