



INSTITUTO POLITÉCNICO NACIONAL  
UNIDAD PROFESIONAL INTERDISCIPLINARIA CAMPUS  
ZACATECAS



**Materia:**

Programación Orientada a Objetos

**Grupo:**

2CM1

**Docente:**

Roberto Oswaldo Cruz Leija

**Investigación 01:**

Diferencias entre ArrayList, LinkedList y List

**Alumno:**

Angel Isaac Saldivar Delgado

**Lugar:**

Zacatecas, Zac

**Fecha de Entrega:**

Jueves 24 de Octubre del 2019

## ***ArrayList***

Esta clase hereda de la clase genérica AbstractList<E>, misma clase que hereda de AbstractCollection<E> (clase genérica también) y ésta última clase hereda de la clase Object. Cuenta con 3 sobrecargas de constructores.

Implementa las siguientes interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, List<E>, RandomAccess

## ***LinkedList***

Hereda de AbstractSequentialList<E>, la última clase hereda de AbstractList<E>, que hereda de AbstractCollection<E> y la última hereda de Object.

Implementa las siguientes interfaces:

Serializable, Cloneable, Iterable<E>, Collection<E>, Deque<E>, List<E>, Queue<E>.

## ***List***

A diferencia de las clases mencionadas anteriormente, List no es una clase, es una interfaz. Hereda de las interfaces Collection<E> e Iterable<E>.

Ahora bien, ¿Cuáles son las diferencias entre cada una?

ArrayList es un arreglo que cambia su tamaño de forma dinámica, LinkedList usa internamente una lista doblemente ligada y List es una colección ordenada de objetos, y las primeras dos clases heredan de List<E>.

LinkedList tiene una desventaja en comparación a ArrayList, que es la forma en cómo se recorren los datos que contienen. La forma de acceder es de forma secuencial, mientras que en ArrayList puede ser desde cualquier posición.

Al usar un objeto de tipo LinkedList, hay que tener en cuenta el uso extra de memoria que se va a usar, ya que tiene apuntadores que apuntan a un nodo anterior y siguiente. En cambio, al usar ArrayList hay que tener en cuenta los costos que hay cuando queremos añadir o remover elementos.

Ambas clases nos sirven para tener colecciones de datos, sólo hay que tener en cuenta las ventajas que tiene cada uno y ver cuál de las dos se nos acomoda mejor al problema que queremos resolver.

## ***Bibliografía***

- <https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html>
- <https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>
- <https://docs.oracle.com/javase/8/docs/api/java/util/List.html>
- <http://www.enrique7mc.com/2016/07/diferencia-entre-arraylist-y-linkedlist/>
- <https://www.geeksforgeeks.org/list-interface-java-examples/>