**PRÁCTICA 15 | LISTAS SIMPLES LIGADAS**

# Introducción

Ahora toca mejorar dos estructuras de datos muy útiles, que fueron programadas en prácticas anteriores: **Pila** y **Cola**. Con esta mejora utilizando listas ligadas en lugar de arreglos, ya no se tiene la restricción de definir un tamaño específico antes de crear cada una respectivamente. La implementación de éstas clases mejoradas, pienso, podría ayudar a simplificar código en algún otro programa.

Teniendo una vista rápida de los métodos que conlleva cada estructura, pienso que esta práctica será bastante sencilla (incluso el tiempo de realización lo indica), tal vez haya un poco más de complejidad para codificar **ColaLista**, pero eso se verá sobre la marcha.

# Desarrollo

Como fue indicado en las instrucciones de la práctica, se crearon dos clases en el proyecto: **PilaLista** y **ColaLista**, ambas agregadas al paquete **edlineal**.

Realmente sí fue bastante sencillo implementar cada clase, e incluso tuvieron menos atributos a comparación de las anteriores versiones de Pila y Cola que se manejaban con un arreglo (memoria estática). Los métodos de ambas hacen uso de otros ya creados en la clase **ListaLigada**, que es la encargada de almacenar los elementos que se ingresan a la PilaLista y ColaLista, facilitando totalmente todo el trabajo.

Cabe mencionar que se necesitó implementar un método de la clase **ListaLigada** para poder imprimir los elementos en orden inverso, que es el que manifiesta una Pila.

A continuación, se indican los archivos creados y modificados junto con sus métodos:

## PilaLista.java (paquete edlineal).

* public PilaLista().
* public boolean vacia().
* public boolean llena().
* public boolean push(Object valor).
* public Object pop().
* public void imprimir().
* public Object verTope().

## ColaLista.java (paquete edlineal).

* public ColaLista().
* public boolean vacia().
* public boolean llena().
* public boolean agregar(Object valor).
* public Object eliminar().
* public void imprimir().

## ListaLigada.java (paquete edlineal).

* private Pila getPilaElementos().
* public boolean insertarInicio(Object valor).
* public Object eliminarInicio().
* public boolean agregar(Object valor).
* public Object eliminarTope().
* public ListaLigada().
* public void imprimirR().

Los métodos subrayados a excepción de “imprimirR” recibieron una pequeña modificación: les fue agregado un contador de elementos contenidos en la lista. Para el caso de imprimirR, fue implementado completamente el método.

## PruebaPilaLista.java (paquete interfaces).

La clase anterior es donde se encuentran las pruebas de la clase “PilaLista”.

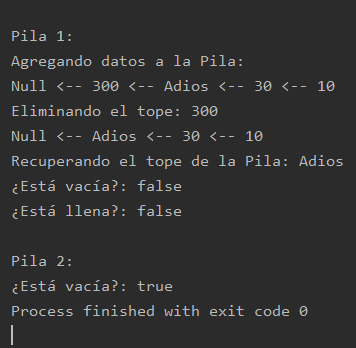
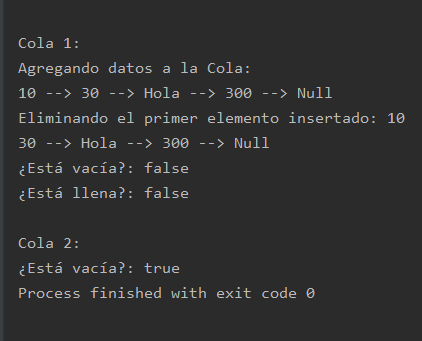
## PruebaColaLista.java (paquete intefaces).

La clase anterior es donde se encuentran las pruebas de la clase “ColaLista”.

**Para mayor información sobre los métodos y la clases creadas vaya a las siguientes ruta dentro del proyecto:**

* **./docs/edlineal/PilaLista.html**
* **./docs/edlineal/ColaLista.html**
* **./docs/edlineal/ListaLigada.html**

# Capturas del programa funcionando



# Conclusiones

Las pilas y colas ya eran estructuras de datos muy útiles, ahora sin necesidad de definir un tamaño específico y limitar su almacenamiento de datos a la memoria de la máquina virtual de Java, las hace aún mejores.