



Universidad Autónoma de Zacatecas

Unidad Académica de Ingeniería Eléctrica

Programa Académico de Ingeniería de Software.

Nombre de la Práctica Listas simplemente ligadas.

Numero de Práctica 15

Nombre de la carrera Ingeniería de Software

Nombre de la materia Lab. Estructuras de Datos

Nombre del alumno Jesús Manuel Juárez Pasillas

Nombre del docente Aldonso Becerra Sánchez

Fecha: 23/09/2021

Práctica 15: Listas simplemente ligadas.

Introducción:

Las listas ligadas nos permiten almacenar una gran cantidad de elementos, pero en cambio con los arreglos estas no es necesario definirles la cantidad de elementos que se estarán almacenando debido a que se utiliza la memoria dinámica. Aunque la forma de acceder a los datos es un poco mas tardado ya que los datos no cuentan con un índice el cual nos diga en que posición esta un elemento si no que tenemos que recorrer la lista y encontrar el elemento deseado.

Desarrollo:

Lo que se pide en esta practica es hacer las clases de pila y cola, pero esta vez utilizando listas ligadas. Aunque serán listas ligadas las que almacenen los datos las clases de ListaPila tendrá el mismo funcionamiento que la clase ArregloPila que utiliza arreglos y así mismo con la Cola.

Para esto se crearon 2 clases dentro del paquete “estructuraslineales”, las cuales fueron ListaPila y ListaCola, ambas clases implementaron la interface de LoteDatos con la que se definieron los métodos que estás clases tendrían.

Los métodos que están clases tendrán son:

- public boolean vacio();
- public boolean lleno();
- public boolean poner(Object elemento);
- public Object quitar();
- public void imprimir();
- public Object verTope();

Para cada clase algunos métodos cambiaran en cuanto a funcionalidad. Los métodos que no cambiaran en cuanto a funcionalidad son los de vacio, lleno, poner e imprimir los cuales indican si la pila o la cola están vacías, llenas (en el caso de llenas, este método siempre regresara falso), o solo agrega los datos a la lista ligada o imprimen la lista ligada que contiene los datos. En cuanto a los demás métodos cambian el lugar en el que se quitan los datos o en el que se extrae el dato que se quitaría, como por ejemplo en el caso de la ListaPila el dato que se quitaría sería el ultimo que se agrego y en el caso de la ListaCola el que se quitaría sería el primero que se agregó, para ver el tope es la misma, pero sin quitarlo solo regresarlo.

Nota: Toda la documentación esta agregada en la carpeta “doc” dentro de la carpeta del proyecto (“edylab_2021_15/doc”).

Capturas del programa funcionando:

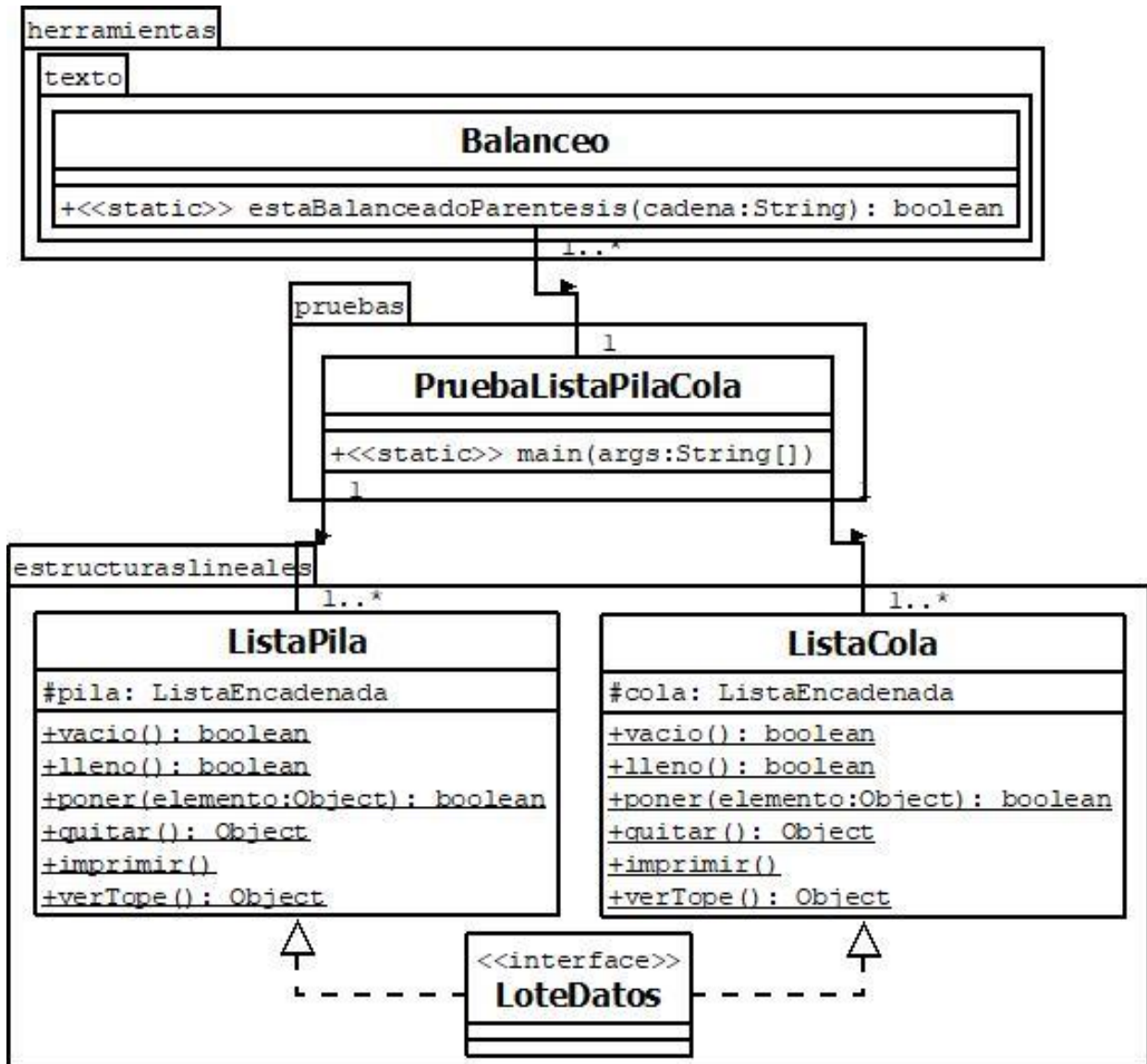
Para realizar las pruebas, estas se hicieron dentro de una sola clase a la cual se le llamo **PruebaListaPilaCola**. En esta clase se probaron los metodos que hicieran lo que una pila o cola dependiendo el caso, tambien se evaluo lo del balanceo de parentesis con la expresion que se dio en el documento de la practica.

```
PRUEBA LISTA PILA:
V -> H -> F -> D -> S -> null
Sacando de pila: S
Ver tope: D
V -> H -> F -> D -> null
-----
PRUEBA LISTA COLA:
R -> G -> D -> S -> X -> null
Eliminando: R
Eliminando: G
Ver tope: D
D -> S -> X -> null
D -> S -> X -> F -> D -> null
-----
PRUEBA ACTIVIDAD 4:
Prueba balanceo de parentesis: true
Ingresa el numero de procesos a ingresar: 3
```

```
Proceso:
    nombre = p1
    comando = c1
    archivo = a1
    ruta = r1
    propietario = p1
    tipo = prioridad: 1
-> Proceso:
    nombre = p2
    comando = c2
    archivo = a2
    ruta = r2
    propietario = p2
    tipo = prioridad: 2
-> Proceso:
    nombre = p3
    comando = c3
    archivo = a3
    ruta = r3
    propietario = p3
    tipo = prioridad: 3
-> null
```

Código agregado:

Todas las clases, atributos y métodos que se muestran son nuevos a excepción de la interface LoteDatos.



Pre-evaluación:

Pre-Evaluación para prácticas de Laboratorio de Estructuras de Datos	PRE-EVALUACIÓN DEL ALUMNO
CUMPLE CON LA FUNCIONALIDAD SOLICITADA.	No
DISPONE DE CÓDIGO AUTO-DOCUMENTADO.	Sí
DISPONE DE CÓDIGO DOCUMENTADO A NIVEL DE CLASE Y MÉTODO.	Sí
DISPONE DE INDENTACIÓN CORRECTA.	Sí
CUMPLE LA POO.	Sí
DISPONE DE UNA FORMA FÁCIL DE UTILIZAR EL PROGRAMA PARA EL USUARIO.	Sí
DISPONE DE UN REPORTE CON FORMATO IDC.	Sí
LA INFORMACIÓN DEL REPORTE ESTÁ LIBRE DE ERRORES DE ORTOGRAFÍA.	Sí
SE ENTREGÓ EN TIEMPO Y FORMA LA PRÁCTICA.	Si
INCLUYE LA DOCUMENTACIÓN GENERADA CON JAVADOC.	Sí
INCLUYE EL CÓDIGO AGREGADO EN FORMATO UML.	Sí
INCLUYE LAS CAPTURAS DE PANTALLA DEL PROGRAMA FUNCIONANDO.	Sí
LA PRÁCTICA ESTÁ TOTALMENTE REALIZADA (ESPECIFIQUE EL PORCENTAJE COMPLETADO).	95%
Observaciones:	

Conclusión:

Al utilizar la ListaLigada como forma de guardar los datos nos permite guardar los datos que queramos sin restricciones de cuantos datos son permitidos guardar.

Aplicar las listas ligadas como forma de guardar los datos lo hace mucho más eficiente ya que no es necesario definir cuantos datos se tienen que guardar y es mucho más eficiente ya que solo utiliza los datos que se quieren y no contiene "posiciones" que no tengan nada y solo estén desperdiciando memoria.