**Sharis Barrios 221370**

**HOJA DE TRABAJO 4**

Repositorio

El enlace al repositorio es el siguiente: <https://github.com/sharlisbg/HDT4>. En este repositorio se evidencia el diseño de ADT para pilas(stack), la clase abstracta y con clase de implementación: ArrayList, Vector, y Lista. con interfaz y clase abstracta y de pilas. Así también, los patrones de diseño Factory y Singleton.

Diagrama UML:

A continuación, se detalla el diagrama UML de clases, que evidencia la relación entre todas las clases que se emplearon para la realización de este ejercicio. La fuente de código y estructura fue tomada del libro Java Structures, ejercicios hechos en clase y hojas de trabajo anteriores.

Diagrama

Descripción generada automáticamente

Pruebas unitarias JUnit:

Se hicieron pruebas unitarias con cada tipo de pila, se introdujo una expresión infix y se tuvo el resultado como expresión postfix.

Texto

Descripción generada automáticamente

Funcionamiento del programa:

A continuación, se evidencia el funcionamiento del programa y la lectura de expresiones infix que nos devuelve un resultado calculado de manera postfix. Estas expresiones se encuentran en un archivo plano llamado datos.txt.

* Arreglo:

Texto

Descripción generada automáticamente

* Vector:

Texto

Descripción generada automáticamente

* Lista Simple:

Texto

Descripción generada automáticamente

* Lista Doble:

Texto

Descripción generada automáticamente

* Excepción con Arreglo:

Texto

Descripción generada automáticamente

Pregunta:

Revisando las ventajas / desventajas que hay al utilizar el patrón Singlenton en general, ya que su comportamiento es muy similar a una variable global. ¿Cree que su uso es adecuado en este programa?

**Para este caso es conveniente tener una sola calculadora ya que simula la realidad de la situación. Solamente se cuenta con una calculadora para realizar diversas operaciones, una seguida de la otra. Además, nos ayuda a controlar la instancia que vamos generando y aunque podamos crear más instancias, siempre nos pegamos a un único objeto. Y en este caso es conveniente implementarla con el Factory ya que también tenemos 4 diferentes tipos de pilas que podemos implementar dentro de la calculadora.**