****

**Universidad Estatal a Distancia**

**Diplomado en Informática**

**Cátedra de Tecnología de Sistemas**

**Escuela de Ciencias Exactas y Naturales**

**Tarea #2**

**Estructuras de Datos**

**Agosto 2015**

Olman A Rojas Espinoza

INTRODUCCION

En la siguiente tarea se va analizar un pequeño algoritmo para la demostración del uso de pilas en el lenguaje de programación JAVA.

Java contiene por defecto una librería que se llama Stack, esta librería contiene métodos sencillos para agregar o eliminar elementos de la pila.

Sin embargo no se pueden eliminar elementos a la mitad del camino o bien escoger un elemento en particular para eliminarlo, o bien remplazarlo, lo único que se puede hacer es eliminar elementos que se encuentran en la cima de una pila.

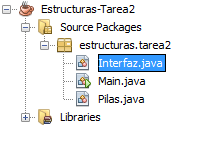
Cuando se quieran eliminar elementos de una pila, primero hay que comprobar que la pila contenga elementos o en su defecto que no se encuentre vacía, ya que si esto ocurre entonces JAVA despliega un error y la aplicación da error.

Para evitar este tipo de contratiempos se le va a preguntar al algoritmo si la pila se encuentra vacía, si esta es falsa continua con el próximo procedimiento, si no por el contrario despliega un mensaje y no ejecuta el método de borrado o eliminación del tope.

Cabe mencionar que todo el código se encuentra documentado internamente.

DESARROLLO

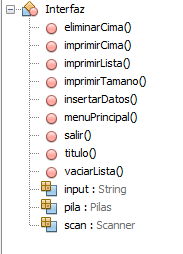
Para empezar enlistamos los archivos con que cuenta la aplicación:



Donde:

**Main.java** es la clase principal que instancia la clase Interfaz.java

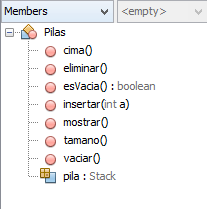
**Interfaz.java** contiene toda la lógica de validación de datos, se asegura que los Datos sean enteros positivos y a la vez se asegura que la pila no esté vacía, además lleva la secuencia y la lógica para controlar el menú, en los siguientes métodos:



Cabe aclarar que Interfaz.java no contiene la lógica para manejar la pila, es simplemente para cuestiones de interacción con el usuario. Interfaz.java lo que hace es instanciar a Pilas.java que esta contiene la lógica para manejar las pilas.

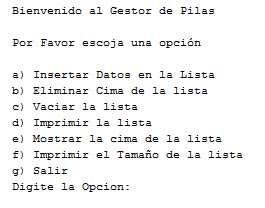
**Pilas.java**

Esta clase contiene los métodos para trabajar las pilas, utiliza la libería Stack de JAVA para todas sus operaciones, Pilas.java contiene los siguientes métodos:

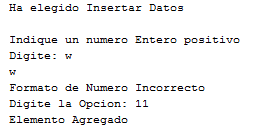


**Prueba de Concepto:**

**Corremos la aplicación:**

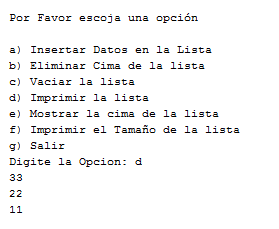


**Escojemos la opción a:**

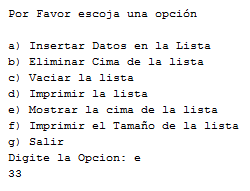


**Se puede comprobar la validación de números y donde este presenta un error al ser agregada una letra.**

**En el siguiente caso hemos agregado 11,22,33, en ese orden, por lo tanto la impresión de la pila se hará alreves:**



**Mostrando la Cima:**



CONCLUSION

Como se pudo observar en el programa anterior, la librería de Stack de JAVA es muy útil si se quieren agregar elementos a una lista sin tener que crear un apuntador o asignar un tamaño exacto a una variable de tipo array.

Sin embargo el problema es el consumo de memoria que esto puede implicar, además que para poder acceder a los elementos estos se deben de sacar de la pila o copiarse en otra pila mientras se encuentra el elemento de interés.

La forma de comportarse de una pila es de tipo LIFO, en otras palabras, el último entrar también es el último en salir.

BIBLIOGRAFIA.

Weiss Mark, A. (2013). *Estructura de Datos en Java.* Madrid España: Pearson Educación S.A.