

## Estructuras de Datos 2018

# ENUNCIADO DEL PROYECTO (VERSIÓN FINAL)

### Bibliografía:

[GT] Michael Goodrich & Roberto Tamassia. *Data Structures and Algorithms in Java. Fourth Edition*. John Wiley and Sons. 2006. El proyecto debe ser implementado en Java (versión 8).

**A) TDA PILA.** Implementar el *TDA Pila*, para un elemento de tipo genérico E, utilizando una estructura simplemente enlazada manteniendo referencia al elemento tope de la pila. Para ello, respete la interfaz *Stack* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**B) TDA LISTA.** Implementar el *TDA Lista* iterable, para un elemento de tipo genérico E, utilizando una estructura simplemente enlazada sin centinelas. Para realizar tal implementación deberá respetar la interfaz *PositionList* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**C) TDA ÁRBOL BINARIO** Defina todas las estructuras de datos necesarias para implementar árboles binarios (AB) utilizando referencias. El tipo del rótulo es de tipo genérico E. Para realizar tal implementación deberá respetar la interfaz *BinaryTree* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**D) TDA MAPEO** Programar todas las interfaces y clases necesarias para implementar el TDA Mapeo que almacene pares (clave, valor), donde el tipo K de las claves es genérico y el tipo V de los valores también lo es, utilizando una Tabla Hash Abierta. Respetar la interface *Map* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**E) Crear un programa** que permita introducir y evaluar expresiones matemáticas con operadores binarios. El programa deberá soportar la definición y almacenamiento de variables y presentará un menú que permita realizar las siguientes operaciones:

1. Definir una variable y asignarle un valor numérico. Para el almacenamiento y manejo de variables deberá utilizar el TDAMapeo implementado en el inciso D. El nombre de las variables puede ser cualquier cadena de caracteres, considere que no pueden existir dos variables con el mismo nombre.

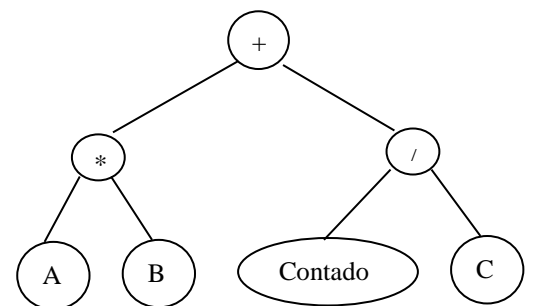
2. Mostrar las variables definidas junto con el valor numérico asociado a cada una.

3. Introducir por teclado una expresión matemática en notación infija, totalmente parentizada y sin precedencia de operadores. Los operadores reconocidos serán los siguientes: ^ (elevado a), \* (multiplicación), / (división), + (suma) y - (resta); y los operandos serán las variables definidas en el punto 1. Los espacios en blanco no se tendrán en cuenta.

Ejemplo: Si se crean las siguientes variables:

- $A \leftarrow 10$
- $\text{Contador} \leftarrow 1$
- $B \leftarrow 2.5$
- $C \leftarrow 4$

Una posible expresión sería:  $((A * B) + (\text{Contador}/C))$



4. Visualizar la expresión introducida en notación prefija. Ejemplo: + \* A B / Contador C

5. Visualizar la expresión introducida en notación infija. Ejemplo: ((A \* B) + (Contador / C))

6. Visualizar la expresión introducida en notación postfija. Ejemplo: A B \* Contador C / +

7. Evaluar la expresión y mostrar el resultado en pantalla. Ejemplo:  $((A * B) + (\text{Contador} / C)) = 25.25$  utilizando recursividad.

8. Evaluar la expresión utilizando una pila y mostrar el resultado en pantalla. (Ayuda: ver aplicaciones de pilas y colas en la clase teórica 5 de la materia).

9. Reemplazar término, deberá evaluar las sub-expresiones cuyos operadores tienen altura 1, crear una nueva variable (solicitando el nombre al usuario), asignarle el resultado y reemplazar el subárbol correspondiente con la variable resultado. Si la raíz tiene altura 1, el árbol no deberá ser modificado.

**10. Mostrar información del árbol binario:**

- Altura del árbol.
- Número de hojas del árbol.
- Número de nodos del árbol.
- Número de nodos internos del árbol.

**En los incisos que considere necesitar una estructura auxiliar, podrá utilizar el TDA Pila o el TDA Lista implementada en los incisos A y B.**

**Implementación de los Tipos de Datos Abstractos**

Implemente los TDA en paquetes diferentes. Nombrelos TDAPila, TDALista, TDAArbolBinario y TDAMapeo respectivamente.

**Programa de prueba:**

Se deberá confeccionar un programa de prueba. El programa de prueba **deberá ofrecer una interfaz gráfica** en la cual se provea acceso y muestre los resultados de cada una de las operaciones que se enunciaron en el inciso (E). Podrá diseñar la interfaz gráfica a su gusto, siempre y cuando provea las funcionalidades especificadas en el inciso (E).

**Documentación:**

- El código entregado deberá estar comentado e indentado adecuadamente.
- Para la corrección, se tendrán en cuenta aspectos relacionados con buenas técnicas de programación, simplicidad y modulación de los algoritmos implementados.
- Manual del usuario: Se debe confeccionar un manual de usuario dirigido principalmente a un usuario no programador. En este manual se debe explicar la funcionalidad del programa y el uso de la interfaz gráfica.
- Documentación javadoc de los TDA implementados y de las clases correspondientes al programa de prueba.
- Minutas del proyecto: incluye cualquier comentario sobre las limitaciones del programa, los alcances de la solución, problemas en el desarrollo del proyecto o cualquier comentario que crea importante.

**Importante:** Los comentarios hechos en persona informalmente a los ayudantes, asistente o profesor no tienen validez documental.

---

**ENTREGA DEL PROYECTO**

---

**FECHA DE ENTREGA: 29 DE MAYO DE 2018 (23:59 hs.)****FECHA DE REENTREGA: 26 DE JUNIO DE 2018 (23:59 hs.)****CONDICIONES DE LA ENTREGA:**

Las condiciones de entrega son simples, pero estrictas. Se deben entregar por mail en un archivo ZIP o RAR a la dirección [mlg@cs.uns.edu.ar](mailto:mlg@cs.uns.edu.ar) de acuerdo a las siguientes indicaciones:

- El archivo no podrá enviarse por mail como archivo adjunto, ya que Gmail filtra los archivos .jar, de modo que deberá subir el archivo a un repositorio (Google Drive, Dropbox, etc.) y enviar el link de descarga por mail.
- El archivo a entregar será de formato comprimido ZIP o RAR con el nombre con el nombre Proyecto-ED-2018-Libreta1-Libreta2.zip (o rar), donde Libreta1 y Libreta2 será reemplazado por los números de libreta de los alumnos que entregan el proyecto.
- El subject del mail será: ED-2018: Proyecto- Libreta1-Libreta2.
- En el cuerpo del mail deberán figurar la comisión, los nombres completos y números de libreta de los alumnos que entregan el proyecto.
- El archivo ZIP o RAR entregado contendrá las siguientes carpetas:
  - **En una carpeta denominada Fuentes** deben figurar los archivos correspondientes a las clases implementadas (todos los archivos .java, imágenes, etc). El código fuente debe ser documentado de acuerdo a la herramienta JavaDoc. Entregar los archivos .java de los TDA implementados y programa de prueba organizados en carpetas.
  - **En una carpeta denominada Jar** se debe guardar el archivo .jar desde el cual se ejecutará el proyecto. Si utilizó jar externos para implementar el proyecto deberá incluirlos.
  - **En una carpeta denominada Documentación** debe incluir el **Manual de Usuario**, las **Minutas del Proyecto** y en una carpeta **Javadoc** la documentación generada por la herramienta JavaDoc para el programador.

Como precaución extra, además de enviar el mail de entrega, se puede optar por entregar un CD o Pendrive (con etiqueta indicando nombres completos y libretas de los alumnos) personalmente en la práctica con el mismo contenido que el enviado por mail. Se le recomienda enviar el mail con copia a usted mismo y verificar que efectivamente ha adjuntado el archivo zip o rar al mail. Las mismas condiciones se aplican en el caso de existir un desperfecto en el servidor de mails.

**La entrega deberá realizarse en tiempo y forma.**

La entrega incompleta o fuera de término implicará que el proyecto sea automáticamente **DESAPROBADO**, debiendo ser reentregado en la fecha correspondiente.