

## Estructuras de Datos 2015

# ENUNCIADO DEL PROYECTO

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

**B) TDA MAPEO** Programar todas las interfaces y clases necesarias para implementar el TDA Mapeo que almacene pares (clave, valor) <K, V> genéricos, utilizando una Tabla Hash Abierta. Respetar la interface *Map* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**C) TDA GRAFO** Programar todas las interfaces y clases necesarias para implementar el TDA Grafo <E> genérico dirigido con vértices pesados utilizando una lista de adyacencias. Respetar la interface *Graph* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**D)** Escriba un programa que utilice los TDA implementados en los incisos anteriores. El programa debe ofrecer las siguientes opciones a realizar sobre un Grafo cuyos vértices tienen rótulos de tipo entero con la estructura descrita en (C). Se puede asumir que el grafo no tiene rótulos de vértices repetidos.

1. Cargar un grafo: Esta opción debe crear un grafo vacío.
2. Agregar vértice: Esta opción permite insertar un vértice especificando su rótulo R.
3. Agregar Arco: Esta opción permite insertar un arco especificando el vértice de origen y el vértice destino.
4. Eliminar Vértice: Esta opción permite eliminar un vértice del grafo especificando su rótulo. Esta opción eliminará también del grafo todos los arcos existentes cuyo origen o destino sea el vértice a eliminar.
5. Mostrar los rótulos de los vértices al recorrer el grafo en profundidad (Depth-first search).
6. Mostrar los rótulos de los vértices al recorrer el grafo en anchura (Breadth-first search).
7. Dados dos números enteros A y B, mostrar el camino más corto del vértice con rótulo A al vértice con rótulo B. En caso que no exista en el grafo un vértice con rótulo A y/o un vértice con rótulo B se debe mostrar un mensaje acorde.
8. Dados dos números enteros A y B, mostrar todos los caminos del vértice con rótulo A al vértice con rótulo B. En caso que no exista en el grafo un vértice con rótulo A y/o un vértice con rótulo B se debe mostrar un mensaje acorde.
9. Dados dos números enteros A y B, eliminar del grafo todos los vértices que forman el camino más corto desde el vértice con rótulo A al vértice con rótulo B (sin eliminar los vértices con rótulo A y B). En caso que no exista en el grafo un vértice con rótulo A y/o un vértice con rótulo B se debe mostrar un mensaje acorde.

### 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 (D).

### 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 al profesor que toma el examen no tienen validez documental.

---

**ENTREGA DEL PROYECTO**

---

**FECHA DE ENTREGA: 16 DE JUNIO DE 2015 (23:59 hs.)****FECHA DE REENTREGA: 30 DE JUNIO DE 2015 (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 [dcic.ed@gmail.com](mailto:dcic.ed@gmail.com) de acuerdo a las siguientes indicaciones:

- El archivo a entregar será de formato comprimido ZIP o RAR con el nombre con el nombre Proyecto-ED-2015-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-2015: 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 (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.