

---

## Estructuras de Datos 2023

# ENUNCIADO DEL PROYECTO

---

El proyecto debe ser implementado en Java.

### Ejercicios:

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

**B) TDA DICCIONARIO.** Programar todas las interfaces y clases necesarias para implementar el TDA Diccionario que almacene pares (clave, valor) <K, V> genéricos, utilizando una Tabla Hash Abierta. Utilizar la lista programada en el inciso anterior. Respetar la interface *Dictionary* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

**C) TDA COLA CON PRIORIDAD.** Programar todas las interfaces y clases necesarias para implementar el TDA *Cola Con Prioridad* que almacene pares (clave, valor) <K, V> genéricos, utilizando un heap. Respetar la interface *CCP* provista por la cátedra. Documentar su implementación utilizando comentarios Javadoc.

### **D) PROGRAMA:**

Se desea crear un sistema para mantener un registro con números de alumnos (LU) y las notas obtenidas en una determinada materia. En el registro no puede haber un alumno con más de una nota. El sistema deberá proveer las siguientes funcionalidades:

1. Al comenzar, deberá solicitar el nombre de la materia. Este nombre debe ser visible en todo momento.
2. Se debe poder ingresar un alumno (LU) y la nota obtenida.
3. Se debe poder consultar por la nota obtenida por un alumno ingresando su LU.
4. Se debe poder eliminar del registro, dado un LU, a un alumno y su nota asociada.
5. Se debe poder visualizar todos los alumnos y sus notas en el registro.
6. Se debe poder calcular el promedio de notas entre todos los alumnos en el registro.
7. Se debe poder calcular mínima nota. Para esto deberá utilizar la CCP programada en (C).
8. Se debe poder mostrar todos los alumnos y sus notas obtenidas de mayor a menor. Para esto deberá utilizar la CCP programada en (C).
9. Se debe brindar una funcionalidad para obtener todos los alumnos que obtuvieron una nota determinada. Para esto debería utilizar si o si el diccionario programado en (B).
10. Se debe poder consultar por los alumnos aprobados y desaprobados. Un examen se encuentra aprobado si obtuvo una calificación mayor o igual a 6.

### **Consideraciones de diseño:**

- Considere que debe tener correctamente separada la GUI de la Lógica y los TDAs en sí.
  - Deberá tener una clase Programa (o como deseen llamarla) cuyos atributos de instancia sean un String que guarde el nombre de la materia y una Lista de Pares. La clase Par se utilizará para guardar el LU y la nota que obtuvo ese alumno. En esta clase se deben programar las funcionalidades (1-10).
  - La clase Par puede programarla usted mismo o utilizar la provista por Java.
-

---

### Consideraciones:

#### **Implementación de los Tipos de Datos Abstractos**

Implemente los TDA en paquetes diferentes. Nómbralos TDALista, TDADiccionario y TDAColaCP; 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 (D). Podrá diseñar la interfaz gráfica a su gusto, siempre y cuando provea las funcionalidades especificadas 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:**

- Utilice únicamente las interfaces dadas por la cátedra.
- Condición necesaria (pero no suficiente) para aprobar: cada TDA deberá superar su correspondiente test junit.
- Los comentarios hechos en persona informalmente a los ayudantes, asistente o profesor no tienen validez documental.
- Cada comisión contará con un “ayudante tutor” al cual deberán consultarle las dudas del proyecto. Ese ayudante es quién corregirá su entrega.

---

## CONDICIONES DE ENTREGA DEL PROYECTO

---

FECHA DE ENTREGA: 23 DE MAYO DE 2023 (23:59 hs.)

FECHA DE REENTREGA: 22 DE JUNIO DE 2023 (23:59 hs.)

### CONDICIONES DE LA ENTREGA:

Las condiciones de entrega son simples, pero estrictas. Se habilitaría una sección en Moodle llamada "Proyecto". Allí encontrará una tarea donde puede subir su proyecto.

Indicaciones:

- El archivo no podrá enviarse por mail .
- El archivo a entregar será de formato comprimido ZIP o RAR con el nombre con el nombre Proyecto-ED-2023-Libreta1-Libreta2.zip (o rar), donde Libreta1 y Libreta2 será reemplazado por los 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.

**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.