

**Programación II - Trabajo Práctico Integrador**  
**2do. Cuatrimestre 2024**  
**SEGUNDA PARTE**

**Fecha de presentación: 22/10/2024(subido al Campus)**

**Fecha límite de entrega: 06/11/2024 (por el Campus)**

En esta segunda parte deben entregar la implementación, el diagrama de clases actualizado, el análisis de complejidad en donde se pida, y el IREP para cada tipo de datos modelado en su solución. Para poder empezar con la segunda parte deben tener aprobado el diseño presentado en la primera parte.

**Requerimientos técnicos:**

- i- Grupos: **El mismo grupo de la primera parte del TP.** Si hay alguna modificación debe ser aprobada por sus docentes de la comisión.
- ii- Se deben utilizar donde sea conveniente las herramientas de Tecnologías Java que se vieron en la materia. Al menos una vez deben usarse:
  - *StringBuilder*, cuyo uso debe basarse en la necesidad de modificar el string.
  - *Iteradores y Foreach* para recorrer las colecciones de Java
- iii- Se deberá utilizar en el desarrollo del trabajo **herencia y polimorfismo**, y al menos 2 de estos conceptos: **sobreescritura, sobrecarga e interfaces**. Como también, en los casos que corresponda, se deberá implementar **clases/métodos abstractos**.
- iv- **En el informe (documento) se debe explicar donde utilizaron estos conceptos.**
- v- **Escribir el IREP de la representación elegida para la implementación de cada TAD. Debe ser parte de la documentación.**

**Por otro lado, desde la materia se proveerá**

- a) Una Interfaz para que se utilice como base para la implementación de la clase principal Aerolinea, con la explicación de cada método. **NO SE DEBE MODIFICAR.**
  - b) Un código cliente con datos para crear los objetos, **NO SE DEBE MODIFICAR.**
  - c) Una clase de testeo (junit). Será condición necesaria para aprobar esta parte del trabajo que tanto el código cliente como el test se ejecuten sin errores.
- Además de pasar el test de *junit* suministrado junto con el TP, en la corrección se testean los ejercicios con otro junit adicional, por lo que se recomienda que el grupo arme un conjunto propio de testeo acorde a su implementación, antes de entregar el TP. Puede entregarlo también si lo desea.

La entrega se realiza subiendo al Campus el proyecto de Java con su implementación **(Seleccionar solamente los archivos .java)** Se debe subir la documentación con los puntos pedidos previamente por escrito en un archivo Word en lo posible o PDF, junto con el proyecto.

## **Consideraciones importantes para la implementación y la documentación del trabajo:**

La implementación de los TADs debe responder a su diseño presentado en la Primera Parte **teniendo en cuenta las correcciones que se indicaron/indicarán a cada grupo. Además, será condición necesaria para aprobar que se cumpla con:**

- Deberá correr satisfactoriamente con el código cliente entregado
- Deberá pasar satisfactoriamente el test junit proporcionado.
- Deberá aprovechar correctamente las estructuras de datos elegidas.
- El código deberá tener implementado el método **toString** del TAD principal, lo que implica que se deban implementar los toString de los TADs relacionados.
- Se deberá usar herencia, polimorfismo y abstracción.

---

## **Algunas modificaciones y aclaraciones al enunciado de la primera parte que se utilizaron para crear a Interfaz (IAerolinea.java) para facilitar la implementación.**

- El origen y el destino de los vuelos son Aeropuertos a los que llega la Aerolínea. O sea, los que ya tiene registrados.
- Los asientos de los vuelos públicos se considerarán numerados correlativamente empezando con clase Turista y terminando con la clase Ejecutivo en un vuelo Nacional. Se numerarán correlativamente empezando con clase Turista, siguiendo por clase Ejecutiva y terminando con Primera clase en un vuelo Internacional. ACLARACION: las clases nombradas se refieren a las secciones en que se clasifican los asientos de los vuelos.
- Los códigos de los vuelos siguen un formato predefinido como se indica en la interfaz.
- Devolver el total recaudado por todos los viajes a un destino dado. **Debe resolverse en O(1).**

### **En el caso de los Vuelos Privados:**

- Se reunió en una sola operación los puntos 6 y 10 de la parte 1. *VenderVueloPrivado* registra el Vuelo Privado y vende el Jet o los Jets necesarios para trasladar los pasajeros. Si son más de uno, la cantidad de jets sólo se usa para calcular el costo total del viaje. El código del vuelo privado es uno solo. Se da por supuesto que, si se necesitan más vuelos, se registra el vuelo y se conoce cuantos jets son, según la cantidad de pasajeros.
- Los acompañantes se asignan a los asientos del Jet que se numeran correlativamente y no hay secciones.
- Los vuelos privados no sirven refrigerios.

### **Cancelación de Pasajes: Hay dos maneras de implementarlo**

- 1) Se borra el pasaje y se libera el lugar para que pueda comprarlo otro cliente. **Debe resolverse en O(1).**
- 2) Se cancela un pasaje dado el código de pasaje. No se pide que sea en O(1).

### **Se agrega un punto 15**

- Devolver un texto con formato dado con el detalle de un Vuelo en particular. El formato está explicado en la Interfaz.

**Con las observaciones anteriores, se deberá implementar el TP, teniendo en cuenta las siguientes condiciones:**

- Se debe poder imprimir la Aerolínea completa mostrando sus datos en formato adecuado para poder comprenderlos. Se espera visualizar los clientes, los aeropuertos y los vuelos según el formato dado para el punto 15 anterior. (condiciones para el *toString()* de Aerolínea).
- Tener en cuenta que, en los vuelos públicos, la sección que ocupa un pasajero conociendo el número de asiento que ocupa, está relacionado con el orden en que se numeran los asientos y la cantidad de asientos que tiene cada sección.

***También se deberá entregar en el documento el siguiente análisis de la complejidad:***

Explicar la complejidad lograda y justificar por medio de Álgebra de Órdenes para el punto:

- *Cancelar un pasaje. Según se pide en el caso 1) que debe ser en  $O(1)$ .*

#### **Test (JUnit):**

Se habilitará el archivo de test en el Moodle, junto a este enunciado, de donde deberán descargarlo. **Se avisará en cuanto esté disponible.**

#### **Código Cliente:**

Ya está habilitado junto con este enunciado, en el espacio del Moodle para el TP, de donde deberán descargarlo.

#### **Interfaz:**

Ya está habilitado junto con este enunciado. Se utilizará como Interfaz para crear el TAD Aerolínea.