



DESARROLLO DE ALGORITMOS

Trabajo Práctico Final

1. **Objetivo:** Todos los alumnos deben hacer el TP final **en grupo con un máximo de dos personas** para obtener la cursada de la materia. Deberán defenderlo oralmente.
2. **Modalidad Del Trabajo:** El trabajo debe subirse obligatoriamente y debe estar aprobado en un 60%.
3. **Como presentar el trabajo:** El trabajo se debe subir a una tarea en PEDCO. Se deben entregar los archivos java correspondientes sin errores de compilación. Se deberá defender en forma oral el mismo, presentando el sistema funcionando y respondiendo preguntas de los docentes de la cátedra.
4. **Cómo se arma la nota final en la promoción:** Estará armada por el trabajo presentado, la defensa del mismo, y los parciales y trabajos entregados en la cursada.

5. **Enunciado**

Una aerolínea necesita un sistema para llevar el control de los aviones y rutas de viaje. Desea mantener los datos de los mismos y realizar con estos datos ciertas operaciones.

El sistema mantiene una **estructura de filas y columnas, donde cada fila representa un día de la semana y cada columna una hora del día**. Existen algunas restricciones:

- Cada avión **vuela una sola vez por día**.
- Las horas en que salen son fijas, y corresponden a las horas entre las **08:00 y las 22:00 (ambas inclusive)**.
- **En cada horario solo vuela un avión**.
- Cada **vuelo tiene un número de vuelo único**.

Los datos de cada avión son:

- Identificación
- Modelo
- Cantidad de vuelos
- Cantidad de asientos
- Kilómetros recorridos

Los datos de cada ruta son:

- Número de ruta
- Ciudad Origen
- Ciudad Destino
- Distancia en kilómetros
- Si es internacional o no



El sistema permite importar 3 archivos de texto¹. En uno de ellos está la información de todos los aviones². En el segundo archivo están los datos de las diferentes rutas que tiene asignadas la aerolínea³. En el tercero, los datos de los vuelos para una semana en particular⁴.

Se desea tener un sistema que permita realizar lo siguiente:

1. Cargar los datos de los archivos en las estructuras que correspondan **sin validar los datos de entrada (asuma que son correctos)**
2. Agregar un nuevo avión a la lista de aviones.
 - Verificar que no se permita ingresar un avión cuya identificación ya existe.
 - Verificar que la identificación del avión cumpla con el Estándar internacional de numeración de aeronaves: para el caso de Argentina:

| | | |
|---|-------|---|
|  Argentina | LV | abc (uso general) |
| | LQ | abc (gubernamental) |
| | LV-X | 123 (experimentales) |
| | LV-S | 123 (Aeronave deportiva liviana) |
| | LV-SX | 123 (Aeronave deportiva liviana experimental) |

- Por ejemplo: LV-ZXS o LQ-AAA o LV-X888 son identificadores válidos
3. Agregar un nuevo vuelo al cronograma de vuelos de la semana, solicitando el número de vuelo, el avión, la hora y el día que debe ocurrir, la ruta que debe recorrer y la cantidad de pasajeros. Se deberán hacer los controles correspondientes de acuerdo a todas las restricciones y emitir mensajes al usuario en caso de que no se pueda cumplir con lo solicitado.
 4. Marcar la realización efectiva de un vuelo, es decir, poder determinar cuando el vuelo aterriza en la ciudad destino. En esta opción se deberán actualizar los datos de los aviones de acuerdo al vuelo realizado (cantidad de vuelos y kilómetros recorridos)
 5. Calcular y mostrar el promedio de pasajeros que **efectivamente** volaron en forma **recursiva** (suponiendo que el avión va con sus asientos completos).
 6. Mostrar para un día dado, la lista de vuelos **ordenada por distancia en kilómetros en forma ascendente en un archivo de texto**.

¹ Se entregan tres archivos de muestra.

² Cada renglón del archivo es un avión, y sus datos están separados por ; (punto y coma).

³ Cada renglón del archivo es una ruta, y sus datos están separados por ; (punto y coma).

⁴ Cada renglón del archivo es un vuelo confirmado, y sus datos están separados por ; (punto y coma).



7. Mostrar los datos de un avión dado.
8. Dado un rango formado por dos distancias (solicitar al usuario), devolver en un arreglo todos los vuelos cuya ruta tenga una distancia comprendida en ese rango.
9. Calcular en forma **recursiva la cantidad de horarios** sin vuelos en la semana.
10. Mostrar para cada día de la semana el primer horario en que hay un vuelo internacional. Si no hay, emitir un mensaje adecuado.

Consideraciones

- Se entregan tres archivos a fin de que el sistema pueda importarlos automáticamente y un archivo con código java para ejemplificar el trabajo de lectura de archivos.
- Implementar el/los TDA que considere necesarios para realizar el trabajo.
- Implementar un menú para realizar las operaciones, de manera que el usuario pueda realizar las mismas en cualquier momento y en cualquier orden.
- Realizar todos los controles necesarios, por ejemplo que no se dupliquen aviones, ni vuelos, que no se pueda marcar como realizado un vuelo que ya aterrizó, que no se pueda agregar un vuelo en una hora que ya hay otro, etc.
- Aplicar el algoritmo de ordenamiento que considere apropiado para el inciso 6 (Burbuja, Burbuja Mejorado, inserción, selección o si lo desea, puede utilizar los algoritmos de Divide y vencerás como HeapSort, QuickSort o MergeSort) .