

Se ha contratado a los alumnos de 1dam para desarrollar una aplicación para gestionar los vuelos de un **aeropuerto**.

**ENTORNOS DESARROLLO.** En cuanto al análisis de la aplicación se pide el DC y el Diagrama de CU así como 3 plantillas de CU diferentes, tener en cuenta que existen dos tipos de usuario, el usuario normal y el administrador que puede hacer cambios en el sistema además de las funcionalidades del usuario normal.

**PROGRAMACIÓN.** De cada vuelo se ha de almacenar: un **identificador** de vuelo, el nombre a **aerolínea** responsable del vuelo, el **origen** y **destino** del vuelo así como la **fecha** (dd/mm/aaaa)<sup>1</sup>, **número de pasajeros máximo** y **precio medio** del mismo.

La aplicación debe validar, mediante excepciones, que la fecha tenga un día entre 1 y 30, un mes entre 1 y 12 y un año mayor o igual de 2023 y menor de 2050. (1p)

Hay dos tipos de vuelos: **nacionales** e **internacionales**. Los vuelos nacionales incrementarán su precio un 20% si el vuelo se produce en el mes de julio y un 30% si se produce en agosto. Los vuelos internacionales incrementarán su precio un 25% si el vuelo se produce en julio o agosto. De los vuelos internacionales además, debe almacenarse la lista de países en los que hace escala. Los vuelos internacionales en los que se haga escala tendrán un **descuento** de un 20% por cada escala que hagan.

Se valorará la utilización de las ventajas de la POO, principalmente el uso de la encapsulación de datos, herencia y polimorfismo, así como la escalabilidad de la aplicación y la interoperabilidad.

**Crea una clase que contenga**, al menos, 4 vuelos Nacionales y 4 vuelos Internacionales con valores diferentes en sus atributos, y que estén almacenados en una colección. No obstante, solo hace falta que sea aleatorio el precio medio del mismo. Crear una aplicación, utilizando el proyecto entregado, de forma que respete el **modelo de capas** visto en clase. Se ofrecerá un menú al usuario con las siguientes acciones<sup>2</sup> (controlar si mete una letra (0,5p)):

1. Método que devuelva la lista de vuelos
2. Consulta de los vuelos que vayan de un origen a un destino y cuyo importe esté dentro del rango de precios introducido por el usuario. (0,5p)
3. Añadir vuelo. Permitir añadir ambos tipos de vuelo, pregunta al usuario lo que desea y sus atributos. Controla la fecha mediante excepción. (1p)
4. Añadir escala a un vuelo internacional identificado por su id. (1p)
5. Eliminar vuelos con un origen y un destino determinado, pedir confirmación al usuario para cada vuelo a eliminar. (0,75p)
6. Volcar información a un fichero texto con la información de todos los vuelos. (0,75p)
7. Volcar información a un fichero binario. (0,5p)
8. Cargar información a partir del fichero binario sustituyendo los vuelos que hubiera en RAM. (0,5p)
9. Cargar información a partir del fichero de texto, sustituyendo los vuelos que hubiera en RAM. (0,75p)
10. Listado ordenado de los vuelos que van a un origen y a un destino determinado por precio medio y a igualdad de precio por número de pasajeros (1 p) Si sólo se hace por un criterio puntúa la mitad.

---

<sup>1</sup> Sin proyecto final, LocalDate

<sup>2</sup>La clase constantes debe contener al menos algún String a modo de ejemplo de cómo debería hacerse.

11. Método que devuelva un mapa que contenga para cada origen, la lista de vuelos que salen de dicho origen al resto de destinos. Verificar la salida de dicho Map y aprovechar para obtener el número de vuelos que parten de cada origen. (1,25p).

Resto puntuación: diseño de la aplicación

Desde el paquete UI se mostrará al usuario los listados solicitados así como el resultado de las operaciones realizadas. Los apartados 2,4,5, 6 y 11 puntúan la mitad si no hay streams, lambda y ref. Métodos.

**NOTA IMPORTANTE:** si no has entregado el proyecto final, para obtener un 10 en el módulo de Programación debes hacer los siguientes apartados:

1. Haz las opciones volcado y carga de fichero en formato json de los vuelos nacionales
2. Utiliza una expresión regular para validar la fecha del vuelo que introduce el usuario (excepto para 29 de febrero).
3. Fechas con LocalDate
4. Utiliza la librería Lombok