**Ejercicio 2:**

Una aerolínea requiere de un programa que le permita gestionar los vuelos que ofrece.

Se sabe que un vuelo tiene un código único, un avión asignado, un origen y un destino. Por su parte, un avión tiene un modelo y una lista de asientos disponibles. Además, un asiento tiene un número, letra y precio que se calcula de la siguiente forma:

Según la letra del asiento se tiene un precio base

|  |  |
| --- | --- |
| **Letra** | **Precio Base** |
| A, C, D, F | $ 80 |
| B, E | $ 60 |

Y según el número del asiento se le aplica un factor adicional:

|  |  |
| --- | --- |
| **Numero** | **Factor** |
| menor a 7 | 1.5 |
| entre 7 a 16 | 1.3 |
| mayor a 16 | 1.15 |

De forma tal que, por ejemplo, un asiento **14D** le corresponde un precio de **$104.**

Tener en cuenta las siguientes reglas que se deben cumplir:

* Todo avión debe tener al menos un asiento y en caso un avión sea retirado del sistema todos los asientos asociados deben ser eliminados también.
* No puede existir un vuelo sin aerolínea, pero una aerolínea puede existir sin vuelos asignados.
* El sistema debe permitir que un vuelo sea registrado sin un avión y luego permitir asignar un avión a un vuelo en específico.

Aplicando los conocimientos vistos en clase, resuelve lo siguiente:

1. Realizar el modelo de clases usando UML. (3 puntos)
2. Implementar las clases necesarias para poder representar el caso presentado. (3 puntos)
3. Crear y registrar 3 vuelos cada uno con un avión asignado que tenga al menos 2 asientos. (1.5 puntos)
4. Implementar un método que reciba un modelo de avión y retorne una lista de todos los vuelos que tengan asignado un avión con dicho modelo. Además, mostrar en pantalla toda la información de los vuelos, avión y asientos asociados. (3 puntos)
5. Usar una Excepción para mostrar en consola el mensaje de error “Modelo de avión inválido” en caso se ingrese un modelo de avión que no exista en el punto anterior (D). (1.5 puntos).