

1. (2 puntos) Crea una **estructura de clases** que permita almacenar información de una aerolínea (Airline):
 - a. Nombre la aerolínea
 - b. Los vuelos indexados por el número de vuelo. Por cada vuelo:
 - i. Número de vuelo (entero)
 - ii. Origen
 - iii. Destino
 - iv. Fecha
 - v. Una lista de pasajeros. Por cada pasajero:
 1. NIF (String)
 2. Nombre
 3. Apellidos
 4. Asiento (entero)
 - c. Un conjunto no ordenado de países en los que opera la compañía.
2. (4 puntos) A partir de la estructura de clases anterior, implementa los siguientes métodos **en la clase Airline**:
 - a. Dado un número de vuelo devuelve los pasajeros. Si el vuelo no existe se debe lanzar una excepción.
 - b. Dado un número de vuelo y un NIF, devuelve true si hay un pasajero con ese NIF en el vuelo. Si el vuelo no existe se debe lanzar una excepción.
 - c. Dado un origen, devuelve los vuelos que tienen ese origen ordenados por número de vuelo.
 - d. Dado un número de vuelo y un asiento, devuelve el pasajero que está en ese asiento, si no hay ningún pasajero en el asiento devuelve null. Si el vuelo no existe se debe lanzar una excepción.
3. (2 puntos) Crea los siguientes test unitarios para probar el método del ejercicio 2.a
4. (2 punto - **Para quien tenga SOLO la 3ª evaluación**) Crea un componente AirlineApp que realice lo siguiente:
 - a. Crea una aerolínea hardcodeada (reutiliza la aerolínea del test)
 - b. Pide al usuario un número de vuelo
 - c. Busca en la aerolínea los pasajeros de ese vuelo y muéstranos en pantalla
 - d. En caso de que se produzca alguna excepción, captúrala y muestra un mensaje de error al usuario.
 - e. Asegúrate de que ninguna excepción provoca la finalización abrupta del programa.
5. (2 puntos - **Para quien tenga la 2ª evaluación**) Crea una jerarquía de clases que represente los diferentes tipos de empleados de una cadena de supermercados. Todo empleado tendrá:
 - a. NIF (string)
 - b. Nombre
 - c. Apellidos
 - d. Horas trabajadasExisten tres tipo de empleados:
 1. Presidente. Tendrán como campo específico:

- a. Nombre de la cadena de supermercados.
- 2. Jefe de tienda. Tendrán como campos específicos:
 - a. Tienda que dirige
- 3. Cajero. Tendrá como campos específicos:
 - a. Tienda en la que trabaja
 - b. Número de caja

Se deben implementar los siguientes métodos:

- 1. Un método para añadir horas trabajadas: dado un número de horas, se añaden a las horas trabajadas totales.
- 2. Un método que muestre en pantalla toda la información del empleado, incluyendo el tipo de empleado.

Todo empleado debe ser presidente, jefe de tienda o cajero