



## Laboratorio 1. Temario A.

### Competencias a desarrollar

- Identifica lo que debe hacer el sistema a desarrollar.
- Elabora el análisis y diseño del programa que debe crear a partir de la situación planteada.
- Aplica el patrón de diseño Modelo Vista Controlador en el diseño de su sistema.
- Codifica lo elaborado en el análisis.

### Ejercicio

El gobierno de los Estados Unidos considera que los inmigrantes son un problema para su país. Entre las nacionalidades de las personas deportadas a diario está la guatemalteca, por lo que nuestro país tiene que estar preparado para recibir todos los vuelos que vienen con personas deportadas. Por esta razón, la Dirección General de Migración necesita un sistema que gestione la información de las personas que están llegando diariamente. Se sabe que cada día llegan tres vuelos con personas deportadas.

El sistema debe presentar estadísticas y cálculos sobre las cantidades de personas deportadas que se reciben en los tres vuelos diarios. Para ello, debe modelar los vuelos con la información necesaria para proveer los siguientes datos:

- Porcentaje que representa la cantidad de mujeres, hombres y niños deportados en el vuelo elegido por el usuario.
- Cantidad total de personas que arribaron en los todos los vuelos (esta cantidad es información compartida entre vuelos).
- Muestre de qué grupo (hombres, mujeres, niños) arribaron más personas en el vuelo elegido por el usuario.
- Advertencia (mensaje de preocupación media (☹)) cuando la cantidad de niños que llegaron en el día es mayor a ocho.

Para calcular lo requerido, el sistema creará tres vuelos con información sobre la cantidad de personas en cada grupo, el número de vuelo, su aerolínea y sus respectivos piloto y copiloto. Así mismo, el piloto y copiloto tendrán nombre, edad y horas de vuelo. Finalmente, un vuelo deberá estar compuesto por un avión (del ejercicio #3) y un aeropuerto. El sistema no gestiona aviones ni aeropuertos, por lo que ambos no podrán existir en el sistema sin un vuelo que los contenga.

Un vuelo deberá ofrecer servicios para mostrar sus datos de instancia, así como para proveer la totalidad de sus pasajeros, para devolver un reporte de su estado a torre de control; y para comparar la cantidad de pasajeros de cada grupo que haya entre dos y tres vuelos (según lo que se ingrese). Normalmente, los métodos de una clase que usan instancias de la misma clase como parámetro son métodos de clase, permitiendo que se puedan comparar las instancias sin riesgo de autoreferencias.



Finalmente, los vuelos también deben poder realizar las comparaciones entre cantidades numéricas de pasajeros para un grupo dado. Es decir, un vuelo debe ser suficientemente inteligente para responder qué cantidad de pasajeros del grupo especificado es mayor sin importar si se le proveen vuelos o cantidades numéricas (además del grupo de pasajeros).

Los pilotos deberán poder dar un mensaje de bienvenida a sus pasajeros. También deberán poder entrenar (generar) nuevos pilotos e intercambiar copilotos entre dos vuelos que se le provean. Por último, presentarán un resumen de su información personal cuando se les solicite.

El sistema debe pedir todos los datos al usuario (de los vuelos, los pilotos, los aviones y los aeropuertos). Luego debe ofrecer al usuario un menú que le permita:

- Mostrar cada uno de los datos listados anteriormente.
- Ejecutar las funcionalidades del piloto de un vuelo elegido.
- Ejecutar las funcionalidades del avión de un vuelo elegido.

El sistema termina cuando el usuario lo solicite.

La Dirección General de Migración le ha contratado para que desarrolle el sistema. Como el costo del programa es elevado, le exigen que entregue un documento de texto con los **requisitos funcionales para el sistema, el propósito de cada clase, atributo y método; y un diagrama de clases UML**. Esta documentación es urgente, pues debe trasladarse a los directivos de Migración para aprobar la compra del sistema. Luego de proveerse la documentación se podrá trabajar en el código como tal mientras se procesa el trámite de pago.

### Material a entregar en Canvas

- Documento .pdf con el análisis (documentación requerida).
- Archivo de imagen (.png, .jpg) del diagrama de clases UML.
- Código fuente de la aplicación desarrollada.

### Evaluación

- **(30 puntos)** Análisis:
  - **(5 puntos)** Se listan correctamente todos los requisitos funcionales del sistema a desarrollar.
  - **(5 puntos)** Se describe el propósito de cada una de las clases identificadas. La descripción es lógica y coherente con lo que hace la clase.
  - **(10 puntos)** Se describe el propósito de cada atributo de cada clase. La descripción es suficiente para comprender la razón de la creación del atributo.
  - **(10 puntos)** Se describe el propósito de la creación de cada método de cada clase. La descripción es suficiente para comprender qué acciones realizará cada método. Dicha descripción debe explicar qué recibe y qué devuelve el método.



- 
- **(30 puntos)** Diagrama de Clases UML:
    - **(10 puntos)** La representación de las clases coincide con el estándar del lenguaje UML.
    - **(10 puntos)** Están representados cada uno de los componentes de la clase de manera correcta y de acuerdo con lo descrito en el análisis.
    - **(10 puntos)** Las relaciones entre las clases están correctas.
  
  - **(40 puntos)** Solución al problema y código fuente:
    - **(10 puntos)** El código es fiel al análisis y diseño presentados.
    - **(10 puntos)** El problema se resolvió usando programación orientada a objetos. Se almacenó la información necesaria creando y llenando objetos de las clases identificadas. Se manejan buenas prácticas de programación OOP.
    - **(10 puntos)** La clase que maneja la interacción con el usuario (interfaz) sólo muestra resultados y obtiene datos, no realiza ninguna acción del problema. Las clases que maneja la interfaz despliegan información sólo donde los requerimientos lo especifiquen.
    - **(10 puntos)** El código fuente está suficientemente comentado para comprender qué se hace, incluyendo el encabezado con la información del autor.