

TC2016 - Programación Orientada a Objetos Proyecto # 2

Forma de Trabajo: Individual.

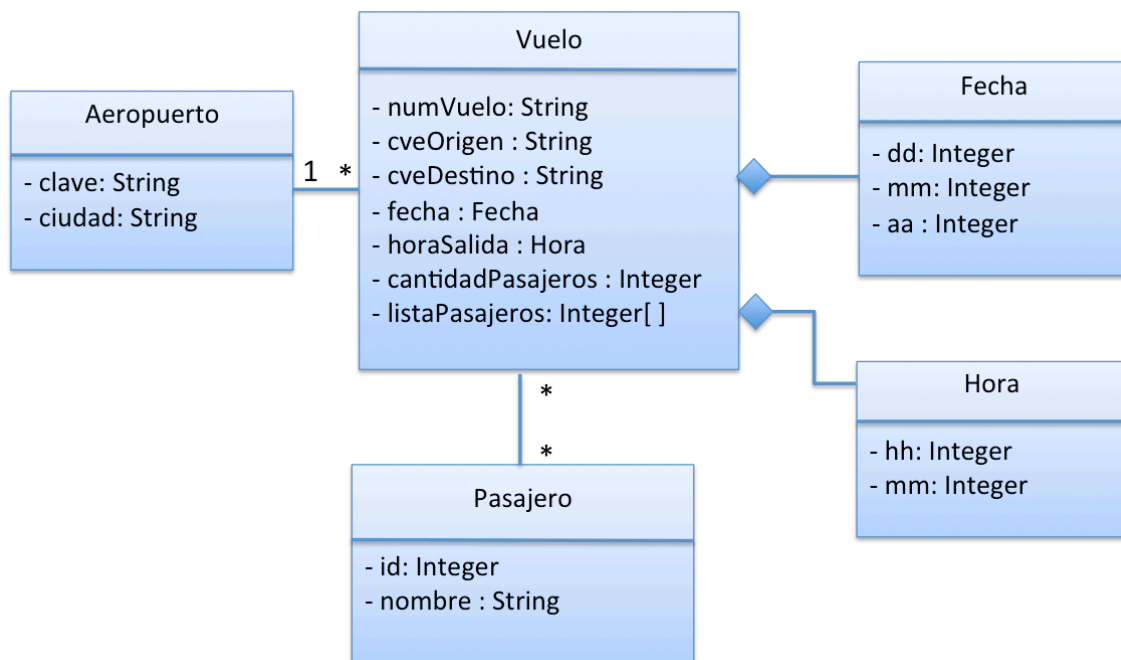
Ponderación: 10 puntos del puntaje final de proyectos.

Programación de clases	Fecha de Entrega: Forma de Entrega:	Viernes 22 de Marzo hasta las 11:59 PM Sube a Blackboard en la sección Assignments en la liga Clases Proyecto2 los archivos .h que contienen las clases del proyecto.
Entrega final	Fecha de Entrega: Forma de Entrega:	Viernes 5 de Abril hasta las 11:59 PM Sube a Blackboard en la sección Assignments en la liga Proyecto2 los archivos .cpp y .h del proyecto.

Descripción del Programa a realizar

Descripción de las clases (la primera entrega incluye solamente las clases, éstas deben compilar para tener los puntos de la primera entrega).

Escribe las clases Fecha, Hora, Aeropuerto, Pasajero y Vuelo de acuerdo con el diagrama de clases siguiente y a las notas que están abajo:



NOTAS:

- Las clases **Aeropuerto**, **Pasajero**, **Fecha** y **Hora** debe tener un constructor default, un constructor con todos los parámetros, métodos de acceso y modificación para todos sus atributos y un método muestra.

- En la clase **Vuelo**, los atributos **cveOrigen** y **cveDestino** se refieren a la clave del aeropuerto origen y destino respectivamente.
- El atributo **listaPasajeros** de la clase **Vuelo** es un arreglo de 10 elementos, en este arreglo se almacena el id de cada uno de los pasajeros que van en este vuelo.
- El atributo **cantidadPasajeros** de la clase **Vuelo** es para almacenar la cantidad de pasajeros que van en el vuelo, es decir, la cantidad de datos que hay en el arreglo **listaPasajeros**.
- La clase **Vuelo** debe tener:
 - **Constructor default** que inicialice el vuelo con la lista de pasajeros vacía y valores default para los demás atributos.
 - Métodos de acceso y modificación para sus atributos **numVuelo**, **cveOrigen**, **cveDestino**, **fecha** y **horaSalida**.
 - Método de acceso para el atributo **listaPasajeros**, debe recibir como parámetro un índice de cuál pasajero (cuál subíndice del arreglo) se quiere acceder.
 - Debe haber un **método** para agregar un pasajero a la lista de pasajeros, éste método debe recibir como parámetro el id del pasajero que se va a agregar al vuelo y el método debe agregarlo al arreglo **listaPasajeros** y ajustar el atributo **cantidadPasajeros**. Es necesario que este método tenga como valor de retorno un valor booleano que indique si se pudo agregar o no el pasajero al vuelo, no se podrá agregar si se duplica o si el vuelo ya está lleno.
 - El atributo **cantidadPasajeros** debe tener solamente método de acceso, porque no es posible modificar la cantidad de pasajeros directamente.

Descripción del proyecto

1. Escribe ahora un programa que tenga arreglos de objetos de tipo Aeropuerto (con un máximo de 5 aeropuertos), Pasajero (con un máximo de 30 pasajeros) y Vuelo (con un máximo de 10 vuelos).
2. Carga los datos para el arreglo de Aeropuertos de un archivo de texto que contiene en cada renglón la clave del aeropuerto y el nombre del aeropuerto (hay un archivo de ejemplo disponible en Blackboard).
3. Carga los datos para el arreglo de Pasajeros de un archivo de texto que contiene en cada renglón el id de Pasajeros y el nombre del pasajero (hay un archivo de ejemplo disponible en Blackboard).
4. Pide al usuario que teclee los datos para cada uno de los vuelos disponibles, en este momento se dará de alta los vuelos sin pasajeros, por lo que la cantidad de pasajeros será de 0.
5. Tu programa deberá tener un menú con las siguientes opciones:
 - a. **Dar de alta un pasajero en un vuelo**, al hacerlo deberás de validar que la clave del pasajero y del vuelo exista. Esto significa que si la clave no existe es necesario volverla a pedir hasta que el usuario teclee una clave correcta.
 - b. **Consulta de vuelo por fecha**, deberás pedir al usuario la fecha, validar que sea válida (supón que Febrero siempre tiene 28 días) y mostrar los datos generales (sin pasajeros) de todos los vuelos que **salen** en esa fecha.

- c. **Consulta por número de vuelo**, debes validar que el id del vuelo exista y mostrar en la pantalla los datos del vuelo, así como los nombres de los pasajeros que van en ese vuelo.
- d. **Consulta de vuelos de un aeropuerto**, debes pedir la clave del aeropuerto, validar que exista y mostrar en la pantalla los datos generales (sin pasajeros) de todos los vuelos que **salen** de ese aeropuerto.
- e. **Terminar**.