

Sigla Asignatura	PGY2121	Nombre de la Asignatura	Desarrollo de Aplicaciones de Escritorio	Tiempo	3 horas (aprox)
Experiencia de Aprendizaje N° 1	Programación Orientada a Objetos				
Actividad	Evaluación de Unidad 2				
Nombre del Recurso Didáctico	Evaluación 2				

1. Contexto del problema

Contab es una idea de aplicación de escritorio ideada por dos Ingenieros en Turismo para apoyar a las personas en gestionar sus vuelos y poder juntar puntos y cobrar viajes por ellos (similar a los puntos por fidelidad).

Hace unos días una amiga le contactó por un problema un tanto urgente, un desarrollador de softwares de Contab renunció en malos términos, y cometió un acto delictual con el código fuente: Borrar el trabajo de varios meses y desarrolladores; ya que la empresa no contaba con Repositorios de códigos, SCM u otros software de protección al código fuente, éste se perdió por completo.

Como solución, acuerdan que usted puede recuperar parte del código eliminado, pero ya se encuentran contrarreloj, puesto que el 2 de Noviembre necesita estar a primera hora el código implementado en los servidores.

Aunque no todo está perdido, ya que los archivos de pruebas al código están intactos gracias a un QA que los tenía en su poder y un DBA tenía un formato borrador de las entidades que se necesitaban.

2. Requerimientos

Se requiere que el sistema cumpla con las siguientes funcionalidades generales:

- Crear una versión del proyecto de esta prueba en el servidor de Github Classroom asignado.
- Completar el código, reemplazando donde sea necesario y ejecutando las funcionalidades.
- El código entregado debe compilar
- El código entregado debe entregar las funcionalidades necesarias.

También se requieren las siguiente funcionalidades específicas:

- Cumplir con el diagrama propuesto.
- Cumplir con las pruebas ya programadas.
- Respetar las carpetas que ya se encuentran definidas.
- Respetar principios de la Programación Orientada a Objetos.

Métodos

Se debe poseer:

- Forma de listar en una misma lista todos los aviones y que sea llamada "listaGlobal".
- Utilización de Interfaces y clases abstractas, como también clases finales dentro de la carpeta "Service" y "Models".
- Todos los objetos de modelo dentro de la carpeta "Models" deben tener 2 constructores (a lo menos), accesores, mutadores y sobrecargas necesarias.
- Una clase que se llamará Validaciones que ayudará a realizar todas las validaciones del sistema.
- Control de excepciones en la carpeta "Exceptions".
- Pasar todas las validaciones del sistema.
- El código debe a lo menos, compilar para ser considerado.
- Las clases que listen deben tener las características completas del CRUD (Guardar, Modificar, Listar, Buscar y Eliminar).
- Se recomienda ver los test para comprobar funcionamiento y piezas a evaluar.
- Todos los Vehículos aéreos deben poseer un método que entregue: Nombre de la nave, su ID, fecha del último vuelo y Horas Voladas, además de su dato único.
 - Ejemplo de nave: "El veloz del cielo, #100, ultimo vuelo: 23/05/2022, total de horas: 44".
- Todos los 100 de las naves deben ser automáticamente generados desde el número 10 y no pueden repetirse entre ellas.
- Email debe estar validado.
- Teléfono debe estar validado.
- Rut debe estar validado.

Objetivo General

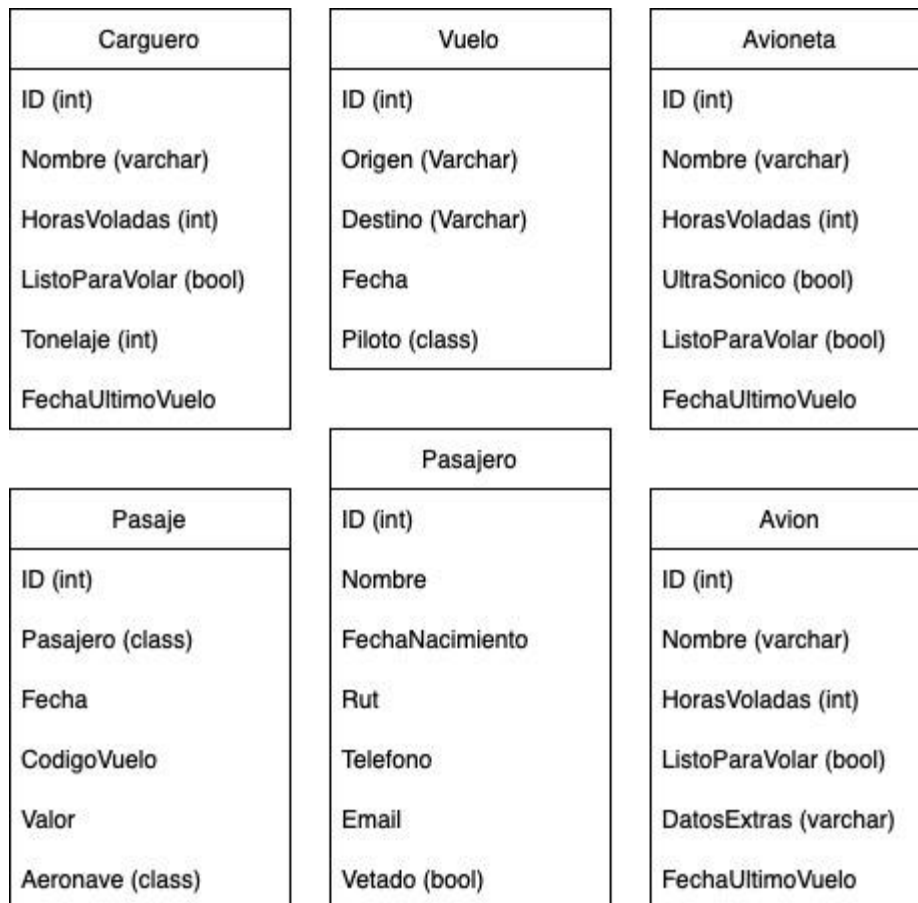
El objetivo final de nuestra prueba es:

- Generar estructura de código a través de models/DTO.
- Generar listas de elementos
- Generar listas generalizadas
- Generar validaciones en un archivo estático
- Pasar validaciones simples de código
- Generar procesos con clases abstractas, finales e interfaces.

Terminal

Para recomendar que para completar el sistema y realice sus propias pruebas, se genere un script en la clase principal/main de la plataforma Java que muestre un ejemplo de como crear pasajeros, suscripciones y sus vuelos, como también el detalle de cada ejecución que van realizando, todo ejecutado desde la implementación.

Diagrama entregado por el DBA:



NOTA:

El código fuente debe subirse al repositorio de Github Classrooms, se recomienda el uso constante de registro y poseer una cantidad apropiada de *commits* al finalizar la entrega.

Se dispone un plazo máximo hasta el día 1ro de Noviembre, a las 23:59 hrs.

A tomar en consideración para enviar:

- Subir su código al repositorio generado por el Github Classrooms.
- No se considerarán *commits* por sobre la hora pactada.