Universidad Tecnológica Nacional
Facultad Regional Avellaneda

1° Parcial Lab II - 2°A - 1°Cuat. 2023

Requerimientos

Nos han contratado para desarrollar el sistema que facilite la administración de los servicios que ofrece una **AEROLÍNEA**. La empresa cuenta con una flota de aviones y ofrece volar a destinos nacionales e internacionales.

La empresa posee un ***administrador***, un ***supervisor*** y tres ***vendedores*** de viajes. Cada uno de estos usuarios (*descargar archivo .json*) designados por la empresa deberán poder registrarse cada uno de ellos con su *correo* y *contraseña* para luego poder realizar las siguientes acciones, según su perfil:

1. Visualizar la lista de viajes disponibles y la información de las mismas.
   - ☐ Deberá visualizarse en el sistema todos los viajes disponibles. Una vez elegido el vuelo, deberá poder visualizar la información básica del viaje, tales como el origen, el destino, hora de partida, precio, listado de pasajeros con información detallada y alguna información más que crea conveniente.

2. Vender un viaje a un pasajero.
   - ☐ Deberá poder vender pasajes, siempre y cuando se pueda satisfacer las necesidades del cliente (por ejemplo, si ofrece o no comida, si tiene servicio de internet, si posee el suficiente espacio, etc.) y si cumple con validaciones básicas: Si el avión está lleno, o si su bodega no tiene más capacidad para recibir las valijas de los pasajeros, si el avión está en vuelo o ya regresó.

3. Consultar estadísticas históricas.
   - ☐ En alguna sección, deberán poder verse los viajes que ya han sido realizados, para poder efectuar cálculos estadísticos sobre ellos, por ejemplo: recaudación, cantidad de pasajeros de cada vuelo, destino más elegido, etc.

4. Crear, modificar y eliminar viajes.
   - ☐ Solo si el perfil es *administrador* se podrán administrar los distintos viajes. Tener en cuenta las verificaciones básicas tales como no eliminar un viaje que se haya

realizado, no modificar un viaje que está en curso, no crear un viaje que tenga fecha de salida en el pasado, etc.

5. Crear, modificar y eliminar pasajeros.
   - ☐ Con el perfil de *vendedor* o *supervisor* se podrán administrar a los pasajeros. Tener en cuenta las verificaciones básicas tales como no eliminar un pasajero que haya realizado un viaje, no modificar un pasajero que está en un curso, no crear un pasajero que tenga el mismo DNI que otro, etc.

6. Crear, modificar y eliminar aeronaves.
   - ☐ Solo si el perfil es *administrador* se podrán administrar los distintos aviones. Tener en cuenta las verificaciones básicas tales como no eliminar un avión que se haya realizado algún viaje, no modificar un avión que está en vuelo, no crear un avión que tenga la misma matrícula que otro, etc.

Nota:
- el perfil de *vendedor* podrá realizar las tareas 1), 2), 3) y 5)
- el perfil de *supervisor* podrá realizar las tareas 3) y 5)
- el perfil de *administrador* podrá realizar la tarea 4) y 6)

Características de la empresa

A nivel ***nacional***, ofrece los siguientes destinos:

- Santa Rosa
- Bariloche
- Corrientes
- Córdoba
- Jujuy
- Mendoza
- Neuquén
- Posadas
- Iguazú
- Salta
- Santiago del Estero
- Trelew
- Tucumán
- Puerto Madryn
- Ushuaia

A nivel ***internacional***, ofrece solo los siguientes destinos:

- Recife (Brasil)
- Roma (Italia)
- Acapulco (México)
- Miami (EE.UU.)

Nota*: todos los vuelos internacionales salen de Buenos Aires.*

Cada una de las aeronaves poseen, al menos, las siguientes características:

- Se identifica a partir de su matrícula (un identificador alfanumérico de 8 dígitos).
- Cantidad de asientos.
- Cantidad de baños.
- Posee servicio de internet o no.
- Ofrece o no comida.
- Capacidad bodega (en kilogramos).

Por otro lado, cada vuelo deberá informar (al menos):

- Ciudad de partida.
- Ciudad de destino.
- Fecha de vuelo.
- Avión en el que se viajará.
- Cantidad de asientos disponibles clase premium.
- Cantidad de asientos disponibles clase turista.
- Costo de cada clase.
- Duración del vuelo.
- Listado de pasajeros.

De cada avión, el **20%** de sus asientos serán para clase *premium*.

La duración del vuelo será calculada de forma aleatoria entre la duración mínima y máxima de cada destino.

Vuelos nacionales: entre 2 y 4 horas.
Vuelos internacionales: entre 8 y 12 horas.

El costo del pasaje se calculará en base a la duración del mismo y el tipo de clase que se ha elegido:

- Cada hora de vuelo costará 50 dólares en destinos nacionales y 100 dólares en destinos internacionales (ambos en categoría *Turista*).
- Clase Turista: Podrá llevar un bolso de mano y despachar solo una valija de hasta 25 kg.
- Clase Premium: Podrá llevar un bolso de mano y despachar hasta dos valijas de hasta 21 kg cada una. Esta clase cuesta un **35%** más de lo que costaría el mismo pasaje para turista.

De los pasajeros se necesita saber (al menos):

- Apellidos y nombres completos.
- Dni.
- Edad.
- Equipajes con los que viaja (puede ser de mano y/o de bodega).

El operador del sistema tiene que poder realizar, al menos, las siguientes acciones:

- Loguearse al sistema con correo y contraseña.
- Visualizar la lista de vuelos disponibles.
- Vender un vuelo (asignar uno o más pasajeros a un vuelo).
- Ver lista de pasajeros.
- Consultar estadísticas históricas.

Consigna

Desarrollar una aplicación de escritorio con **Windows Forms** que resuelva las siguientes necesidades del cliente. Todas las entidades deben realizarse en un proyecto de tipo Librería de clases.

Se deberá poder:

- Acceder a la aplicación ingresando correo y contraseña que serán verificadas deserializando a JSON el archivo usuarios.
- Ver la lista de vuelos con la cantidad de asientos disponibles, su respectivo listado de pasajeros (con información completa).
- Poder consultar por: dni y/o apellidos y/o nombres del pasajero.
- Contar con una barra de información de la aplicación donde figure el nombre del operador conectado y la fecha actual (sin la hora).
- Visualizar fácilmente el estado actual del vuelo (disponible / completo/ etc.).
- Poder venderle un pasaje a un pasajero y asociarlo a un vuelo.
- De los aviones se deberá poder ver toda su información bien detallada.
- Se le deberá mostrar al usuario el costo del vuelo, el costo final bruto por el servicio y el neto a pagar una vez aplicado IVA y las tasas correspondientes.
- Poder visualizar estadísticas históricas:
    - Lista de destinos ordenados por facturación de forma descendente.
    - Lista de pasajeros frecuentes ordenadas por cantidad de vuelos.
    - Ganancias totales y clasificadas por servicio (cabotaje /internacional).
    - Horas de vuelo de cada aeronave.
    - El destino más pedido por los clientes.
    - etc.
- Realizar CRUD sobre: aviones, viajes y pasajeros.
- Serializar / deserializar: usuarios, aviones, viajes y pasajeros.

Criterios de evaluación

Formato de entrega

El examen será recibido, vía *Google Form*, hasta el día **miércoles 17 de mayo** a las 10 hs.

En el formulario de *Google* se pedirá que el alumno registre sus datos personales y la **url** de un repositorio de ***github*** dónde se encontrará ubicado su código.

Dicho repositorio, nombrarlo como *Apellido.Nombre.PrimerParcial*.

El repositorio **debe** configurarse cómo **privado** y agregar cómo colaboradores los usuarios de los docentes:

- ★ Maximiliano Neiner: ***maxineinerutn***
- ★ Facundo Rocha: ***facc15***

Esto último se realiza con el fin de que su código no sea 'copiado' sin su consentimiento.

Recordar que, códigos parecidos / muy parecidos / idénticos implica la reprobación de ambos parciales.

El repositorio debe tener informados varios ***commits*** (desde su creación hasta la fecha de entrega).

IMPORTANTE

La creación del repositorio privado y el posterior agregado de los colaboradores deberá realizarse de inmediato.

Documentación

Se deberá completar la información indicada en el archivo **README.md** que se encuentra en el repositorio.

Para trabajar con este archivo se deberá utilizar el lenguaje de marcado Markdown.

No se corregirá ni revisará ningún parcial que no presente esta documentación completa.

Secciones e información a documentar

- **Título**: Ponerle un nombre a la aplicación.
- **Sobre mí**: Presentarse brevemente. Contar su experiencia programando y lo que significó para vos este trabajo (¿fue un desafío? ¿fue fácil? ¿aprendiste? ¿te divertiste? etc.).
- **Resumen**: Explicar qué hace la aplicación y cómo se usa a grandes rasgos.
- **Diagrama de clases**: Pegar una foto del diagrama de clases correspondiente a la lógica de negocio. Se debe construir con la herramienta del Visual Studio y deberá estar actualizado a la última versión entregada de la solución.
- **Justificación técnica**: Indicar tema a tema (de los temas 01 al 09) dónde se fue aplicando en el código **y por qué se decidió implementarlo de esa forma**. Toda decisión tiene que estar argumentada con razones técnicas que giren alrededor de los pilares de la programación orientada a objetos y cuestiones de mantenibilidad, código limpio, flexibilidad al cambio, experiencia de usuario, accesibilidad, uso seguro, rendimiento y eficiencia.
    - Suma identificar pros y contras, si los tienen en mente.
    - El objetivo es que demuestren que saben lo que hacen y que tomaron decisiones con criterio y no mecanizadas.
    - Si se utilizó alguna biblioteca externa también se deberá justificar la elección.
- **Propuesta de valor agregado para promoción**: En esta sección se explicará y justificará la funcionalidad adicional propuesta para el punto de promoción.

Condiciones mínimas de aprobación

Para alcanzar la aprobación (**nota 4**) se deberán cumplir todas las siguientes pautas:

- Respetar **TODAS** las reglas de estilo de la cátedra y buenas prácticas indicadas en clase. Se corregirán todas las pantallas (visualización) y calidad de código según las exigencias de la cursada.
- Compilar sin errores ni advertencias (sí se admiten sugerencias del IDE).
- Debe resolver **TODAS** las necesidades del cliente (planteadas en la consigna y en el requerimiento) y **NUNCA** tener errores en el tiempo de ejecución.
- Separar de forma física (distintos proyectos) la capa de presentación (interfaz de usuario) de la lógica de negocio (entidades).
- La verificación de ingreso para los usuarios que ingresan al sistema, deben ser realizados ***deserializando*** el archivo usuarios.
- Los datos de uso que no se cargan manualmente deben encontrarse serializados (XML o JSON) así como algunos datos históricos que simulan ejecuciones previas del programa.
- Las clases y sus miembros deberán estar correctamente documentados con la herramienta de documentación **XML** o ser lo suficientemente autodescriptivos (esto último queda a criterio del docente corrector).
- Validar **todos** los ingresos de datos (cuando corresponda) mostrando mensajes claros para el usuario cuando un dato sea inválido.
- Abstraer las entidades y realizar un diseño orientado a objetos. Aplicar todos los temas de los temas 01 al 09 incluido.
    - Al menos dos formularios.
    - Se valorará el uso justificado de formularios **MDI**, pantallas modales y no-modales, menú de opciones.
    - Ninguna entidad se debe comparar por defecto, sino por uno o varios de sus atributos (idealmente por su identificador). Se deberá cambiar el comportamiento de todos los métodos de comparación (método **Equals** y método **GetHashCode**).
    - Todas las entidades sobrescribirán el método **ToString** y retornarán una cadena de texto con los datos que sean necesarios del objeto según el modelo utilizado.
    - Al menos una jerarquía de herencia que aproveche el pilar del **polimorfismo**.
    - Todas las entidades deberán estar correctamente encapsuladas exponiendo sólo sus operaciones y características esenciales, protegiendo el acceso y modificación libre de datos, y ocultando los detalles de la implementación.
    - Al menos una sobrecarga de constructores y una sobrecarga de métodos.
    - Clases, atributos y métodos que no correspondan o trabajen con el estado de una instancia particular deberán ser estáticos.
    - Todos los objetos deberán inicializar su estado con los mínimos valores necesarios para que no exista lugar a fallos en el uso del objeto, no debiendo permitir que se instancien de otra forma.
    - Declaración y uso de al menos dos **enumerados**.
    - Declaración y uso de al menos dos tipos de colecciones genéricas distintas.
    - Uso justificado de al menos una clase abstracta.

Si no se cumplen TODAS las condiciones mínimas de aprobación, no se continuará con la corrección y la nota será un dos (desaprobado).

Condiciones mínimas para promocionar

Para promocionar (**nota 6 o más**) se deberá cumplir todas las condiciones mínimas de aprobación, agregar las funcionalidades descritas más abajo y proponer una nueva funcionalidad en base al contexto del negocio.

La nueva funcionalidad deberá ser agregada a la aplicación siguiendo todos los criterios de calidad y buenas prácticas antes nombrados.

Justificar el valor agregado de su elección en la sección ***"Propuesta de valor agregado"*** del archivo **README.md**.

Además se deberá agregar:

- Tener la posibilidad de cancelar el cierre de la aplicación.
- Herencia gráfica (herencia de formularios).
- Para todos los formularios se debe poder maximizar, minimizar y cambiar el tamaño de la ventana.
    - La posición y/o tamaño de los controles deberá ajustarse con la ventana. *Pista: hay una propiedad específica para esto*.
    - Deberá existir un límite mínimo para ajustar el tamaño que será aquel donde se pierde visibilidad de los controles o se dificulta el trabajo con la aplicación.
- Agregar sobrecarga de operadores, indexadores y los implícitos y explícitos.
- Crear un archivo ***usuarios.log*** que agregue toda la información del usuario que ha accedido a la aplicación, informando la fecha de acceso con formato ***'año-mes-día hora:minutos:segundos'***.
- Realizar los CRUDs serializando y deserializando en:
    - JSON para aviones y usuarios.
    - XML para viajes y pasajeros.
    - A elección del alumno para las estadísticas.

Si no se cumplen TODAS las condiciones mínimas de promoción, se corregirá el parcial pero la nota no podrá superar el cuatro (aprobado sin promoción).

IMPORTANTE

*Una vez cumplidas las condiciones y, en caso que el docente crea necesario, se llamará **en cualquier momento de la cursada** para defender el proyecto de forma **oral** para contar cómo lo resolvieron o que función cumple un determinado módulo.*

*Esta defensa será **presencial**.*

*El resultado de la exposición oral **puede anular la promoción** si se detecta que no conocen en profundidad el proyecto o no entienden lo que hicieron.*

Por el contrario, una muy buena exposición sumará un punto a la calificación.

Habrá casos donde no se sume el punto pero se mantenga la promoción.

Calificación

Una vez que se hayan superado las condiciones de aprobación, se procederá a definir la nota por tema evaluando un uso correcto y bien justificado de cada uno de los temas vistos entre la clase 01 y la 10.

A tener en cuenta
Reglas de estilo, buenas prácticas y calidad de documentación.
Algoritmos y resolución de problemas. Reutilización de código. Elección de tipos de datos. Trabajo con cadenas de texto, fechas, etc.
Uso correcto y justificado de miembros estáticos.
Abstracción. Diseño de clases y objetos (diagrama de clases, atributos, métodos, constructores).
Uso correcto y justificado de sobrecargas de métodos, constructores y operadores.
Uso correcto y justificado de colecciones.
Creatividad en el diseño de formularios. Uso correcto de controles y eventos. Buena experiencia de usuario.
Encapsulamiento y uso correcto y justificado de modificadores de acceso.
Uso correcto y justificado de propiedades y enumerados.
Uso correcto y justificado de jerarquías de herencia, clases abstractas y selladas.
Uso correcto y justificado de polimorfismo.
Uso correcto y justificado de serialización / deserialización de objetos.
Investigación, instalación y uso de una biblioteca externa (paquete Nuget) que resuelva ayude a resolver alguna necesidad de la implementación.
Defensa oral presencial.

Consultas sobre el parcial

Se podrán responder consultas sobre el parcial durante las clases (NO ocupará toda la clase).
De existir la necesidad de hacer alguna consulta en otro momento de la semana, no se garantizará una respuesta veloz ya que la misma quedará sujeta a disponibilidad del docente y/o sus ayudantes.

Todas las preguntas deberán realizarse por Classroom como único medio y sin excepción por la clase PRIMER PARCIAL (en los Comentarios de la clase, NO cómo comentarios privados).