

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	06/12/2018
Nombre:		Docente ⁽²⁾ :	F. Dávila / H. Dillon
División:		Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<input type="checkbox"/> PP <input type="checkbox"/> RPP <input type="checkbox"/> SP <input type="checkbox"/> RSP <input type="checkbox"/> FIN <input checked="" type="checkbox"/> X		


(1) Las instancias validas son: 1^{er} Parcial (PP), Recuperatorio 1^{er} Parcial (RPP), 2^{do} Parcial (SP), Recuperatorio 2^{do} Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

IMPORTANTE:

- **2 (dos) errores en el mismo tema anulan su puntaje.**
- **Cada tema vale 1 (un) punto (Herencia, Generics, Test Unitarios, etc.). La correcta documentación también será evaluada.**
- **Se deberán tener al menos el 60% de los temas de cada parcial bien para lograr la aprobación.**
- **Errores de conceptos de POO anulan el punto.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.AñoCursada. Ej: Pérez.Juan.2018. No se corregirán proyectos que no sea identificable su autor.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.

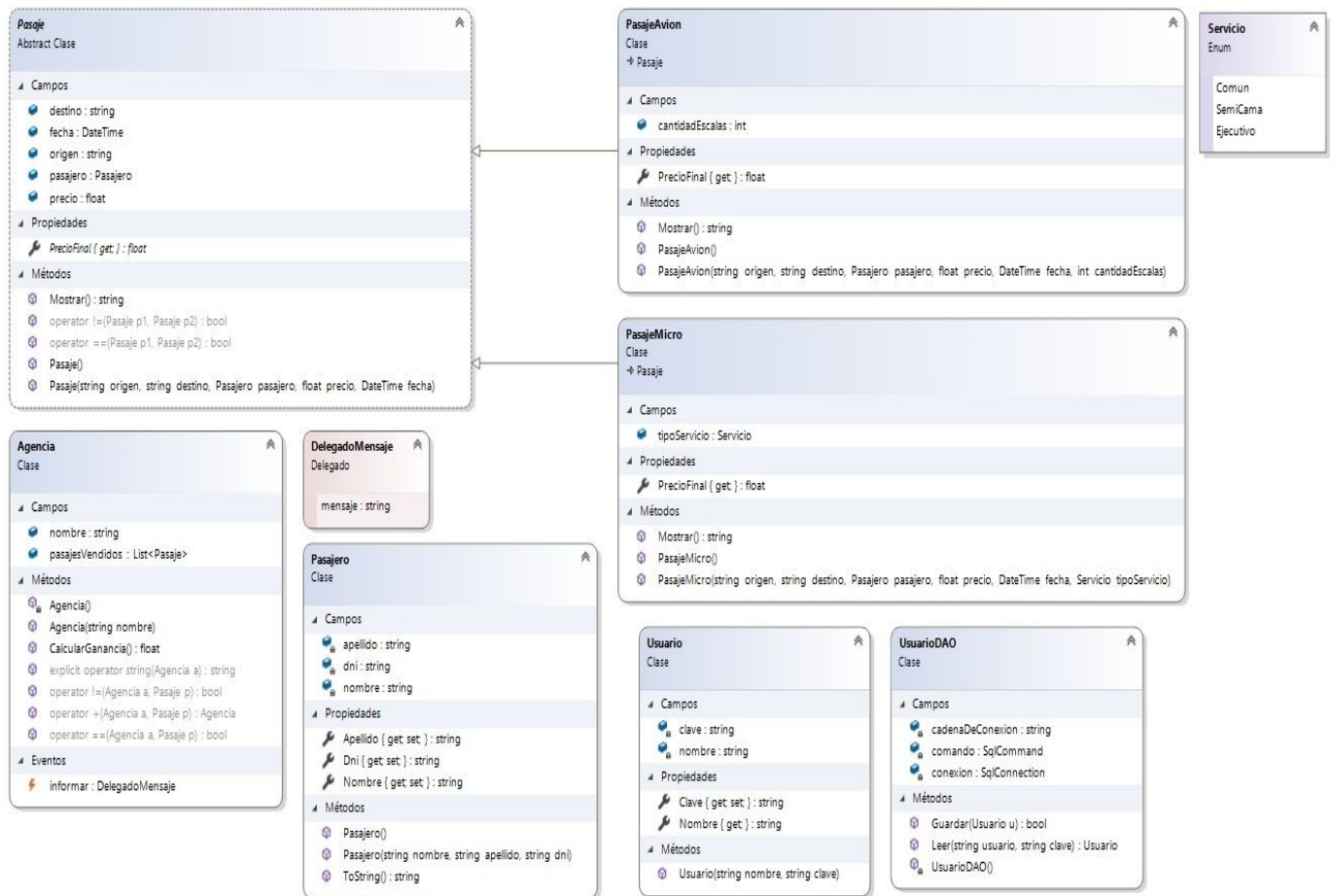
Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre

Apellido.Nombre.AñoCursada.zip y dejar este último en el Escritorio de la máquina. Luego presionar el botón  de la barra superior, colocar un mensaje y apretar **Aceptar**. Luego retirarse del aula y aguardar por la corrección.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 100 MINUTOS.

El examen consiste en una simulación de una venta de pasajes de avión y micro para una agencia de viajes.

1. Restaurar el archivo **final-20181206.bak** otorgado. El mismo creará una base de datos llamada **final-20180802** con una única tabla llamada "Usuarios".
2. Dentro de la Biblioteca de clases llamada Entidades agregar el contenido correspondiente a las siguientes clases:

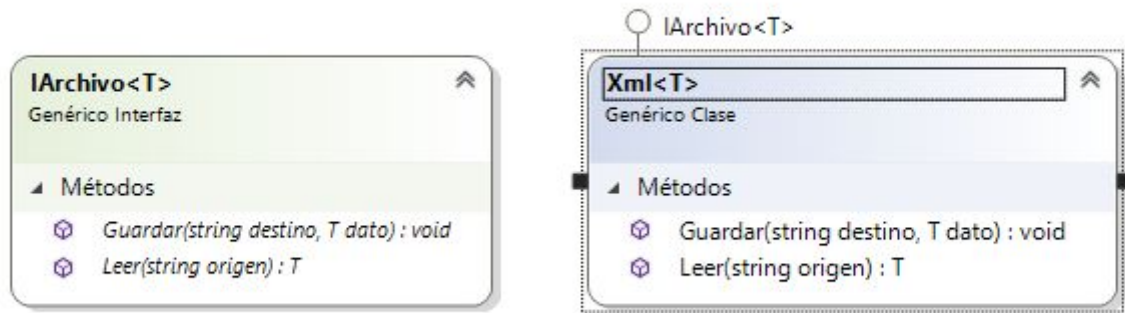


- Sólo final:** Utilizar la teoría de encapsulamiento en **todas** las clases.
- Las clases Usuario y UsuarioDAO serán utilizadas para insertar y recuperar usuarios de la base de datos a través del formulario Login detallado más adelante.
 - Para la clase Usuario sólo se requiere modificar el set de la propiedad Clave donde se deberá validar que la clave proporcionada no sea menor a 8 dígitos, caso contrario se pide lanzar una excepción del tipo "ClaveInvalidaException" con el mensaje "La clave debe contener al menos 8 dígitos!".
- Clase Pasaje
 - Sólo final:** La propiedad PrecioFinal no tendrá implementación dentro de Pasaje.
 - Mostrar tendrá implementación y podrá ser sobrescrito en las clases hijas. Deberá mostrar todos los datos de la clase donde se lo implemente.
 - Sólo final:** Sobrecargar el operador == entre 2 pasajes los cuales serán iguales sólo si tienen el mismo origen, destino, dni del pasajero y fecha.
- Clase PasajeAvion:
 - Mostrar agregará los datos propios.
 - La propiedad PrecioFinal retornará el precio aplicando un descuento del 10% para viajes con una escala, 20% con dos escalas y 30% con más de 2 escalas.
- Clase PasajeMicro:
 - Mostrar agregará los datos propios.
 - La propiedad PrecioFinal retornará el precio aplicando un aumento del 10% para tipo de servicio SemiCama y 20% para Ejecutivo.
- Clase Agencia:
 - Crear un delegado llamado DelegadoMensaje por fuera de la clase con la siguiente firma "void DelegadoMensaje(string mensaje)" y un evento llamado **informar** cuyo tipo será el delegado antes mencionado.
 - La lista de pasajes se deberá inicializar sólo en el constructor privado.
 - El operador == entre Agencia y Pasaje retornará true si el pasaje ya se encuentra en la lista.
 - El operador + agregará un pasaje a la lista siempre y cuando este no se encuentre agregado. En caso de haber podido agregar el pasaje a la lista, invocar al evento **informar** con el

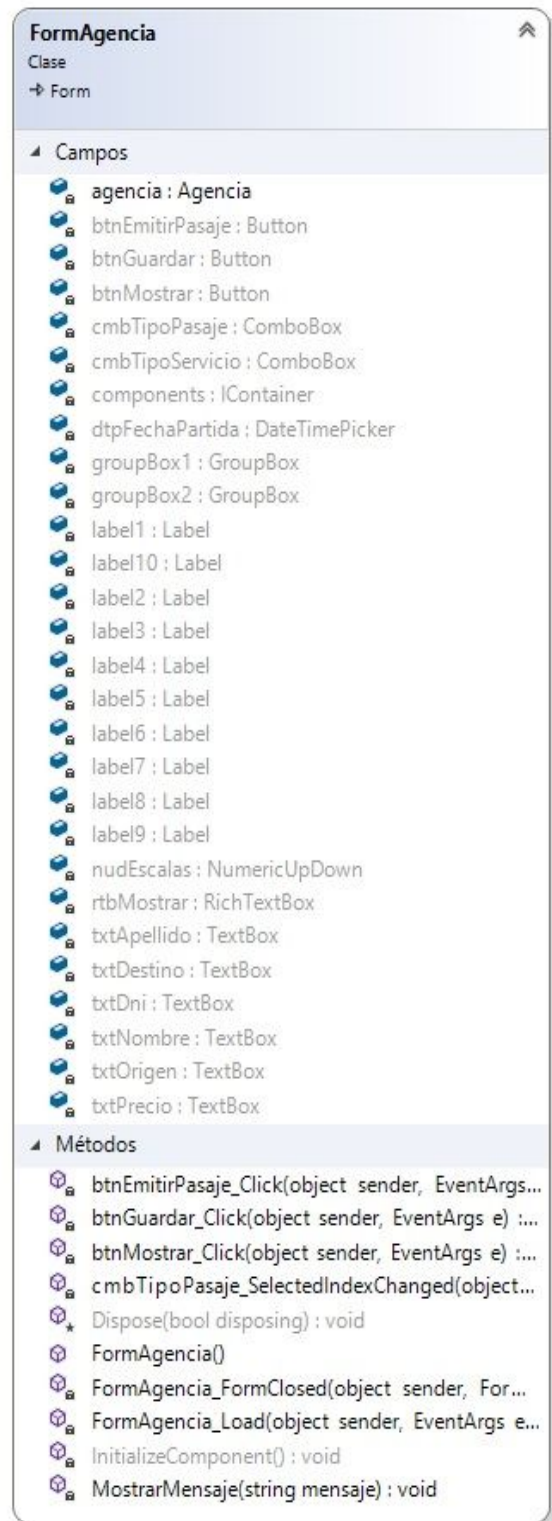
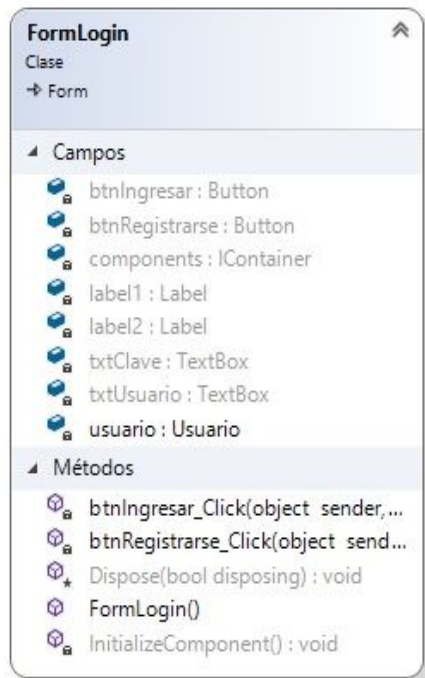
mensaje "Pasaje emitido correctamente.", caso contrario **informar** con el mensaje "El pasaje ya fue emitido con anterioridad."

- v. CalcularGanancia retornará la suma del precio final de todos los pasajes vendidos.
- vi. **Sólo final:** Realizar una conversión explícita a string que retorne todos los datos de la agencia y sus pasajes vendidos.

2. Crear una nueva biblioteca de clases llamada Archivos con las siguientes clases:
 - a. La clase Xml será utilizada para serializar y deserializar un objeto dado.




3. Dentro del formulario **FormLogin** ubicado en el proyecto VistaForm se deberá:
 - a. En el manejador del evento Click del botón Ingresar:
 - i. Agregar el código necesario para recuperar el usuario de la base de datos.
 - ii. Agregar captura de errores.
 - iii. En caso de no existir el usuario ingresado, mostrar el mensaje "Usuario o Clave incorrectos" con un ícono de exclamación.
 - iv. En caso de error, mostrar el mensaje "Error al intentar recuperar el usuario" con un ícono de error.
 - b. En el manejador del evento Click del botón Registrarse debajo del comentario "//Alumno" Agregar el código necesario para :
 - i. Guardar el usuario en la base de datos.
 - ii. Agregar captura de errores.
 - iii. En caso de guardar el usuario correctamente mostrar el mensaje "Usuario registrado".
 - iv. En caso de error por clave inválida, mostrar el mensaje "La clave debe contener al menos 8 dígitos!" con un ícono de error.
 - v. Ante cualquier otro error, mostrar el mensaje "Error al registrar el usuario" con un ícono de error.
4. Dentro del formulario **FormAgencia** ubicado en el proyecto VistaForm se deberá:
 - a. En el manejador del evento Click del botón EmitirPasaje:
 - i. Crear el pasaje con los datos ingresados en el formulario
 - ii. Agregarlo a la lista de pasajes vendidos de la agencia.
 - b. En el manejador del evento Click del botón Guardar:
 - i. Serializar la agencia del formulario en un archivo llamado "Agencia.xml", en caso de error al serializar informar al usuario.
 - c. En el manejador del evento Click del botón Mostrar:
 - i. Mostrar en el RichTextBox rtbMostrar los datos de la agencia.
 - d. Crear un manejador para el evento **informar** de agencia llamado MostrarMensaje el cual sólo mostrará al usuario el mensaje recibido.
 - e. Asociar dicho evento a su manejador en el evento load del formulario.



5. Agregar un test unitario para probar serializar una agencia y al deserializar obtener la misma información.

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y dejar este último en el Escritorio de la máquina.

Luego presionar el botón  de la barra superior, colocar un mensaje y apretar **Aceptar**. Finalmente retirarse del aula y aguardar por la corrección.