

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	22/11/2018
Nombre:		Docente ⁽²⁾ :	F. Dávila / H. Dillon
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<input checked="" type="checkbox"/> PP <input type="checkbox"/> RPP <input type="checkbox"/> SP <input type="checkbox"/> RSP <input type="checkbox"/> FIN		

(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:

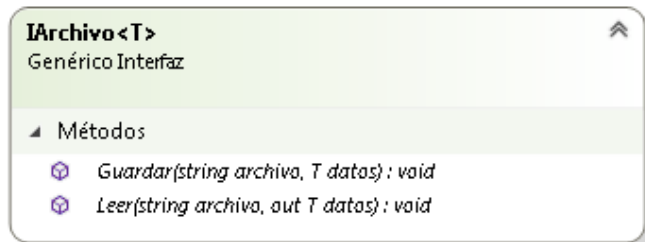
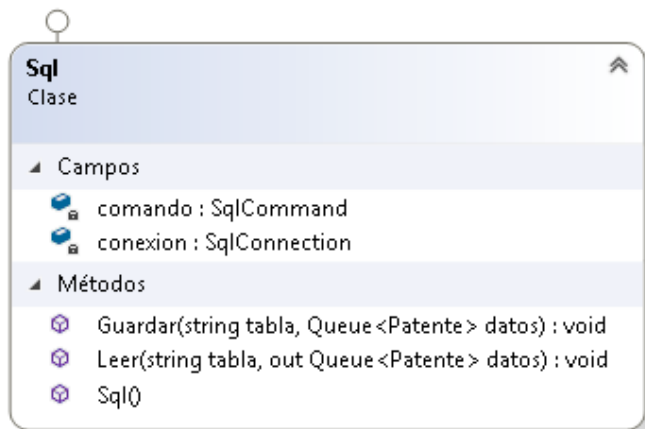
- **2 (dos) errores en el mismo tema anulan su puntaje.**
- **La correcta documentación y reglas de estilo de la cátedra serán evaluadas.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.Departamento. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- **TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.
- Colocar nombre de la clase (en estáticos), **this** o **base** en todos los casos que corresponda.

TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.

1. Partir de la solución entregada. Modificar su nombre con el siguiente formato: [APELLIDO].[NOMBRE]

Archivos

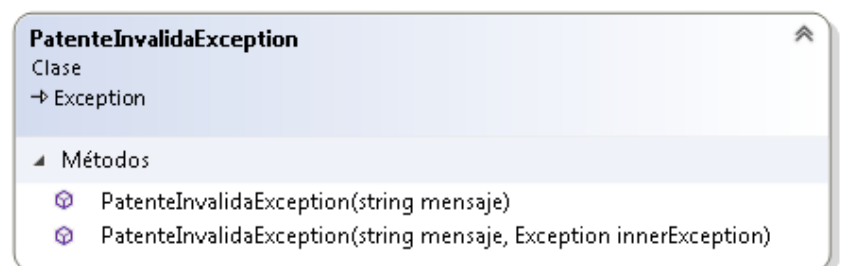
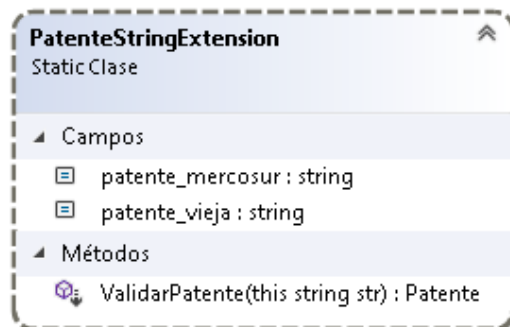
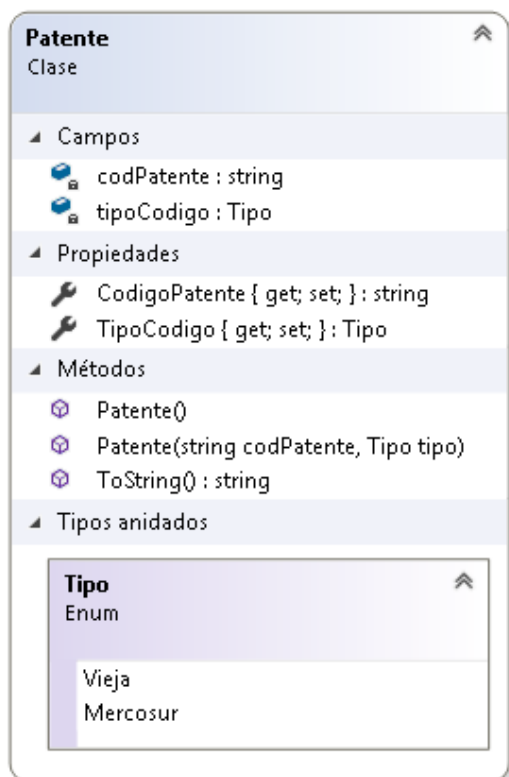
2. Dentro del proyecto **Archivos** se deberá respetar el siguiente esquema:



- Las clases Sql, Xml y Texto implementarán la interfaz IArchivo para acceder y consultar los datos de una base de datos, un archivo serializado en formato Xml y un archivo de texto (provistos junto al examen y ubicados en el escritorio).
- Sql recibirá el nombre de la tabla a consultar (patentes) y una cola con los datos.
- Xml y Texto recibirán el nombre del archivo (**sólo el nombre**) y los datos a leer o guardar.

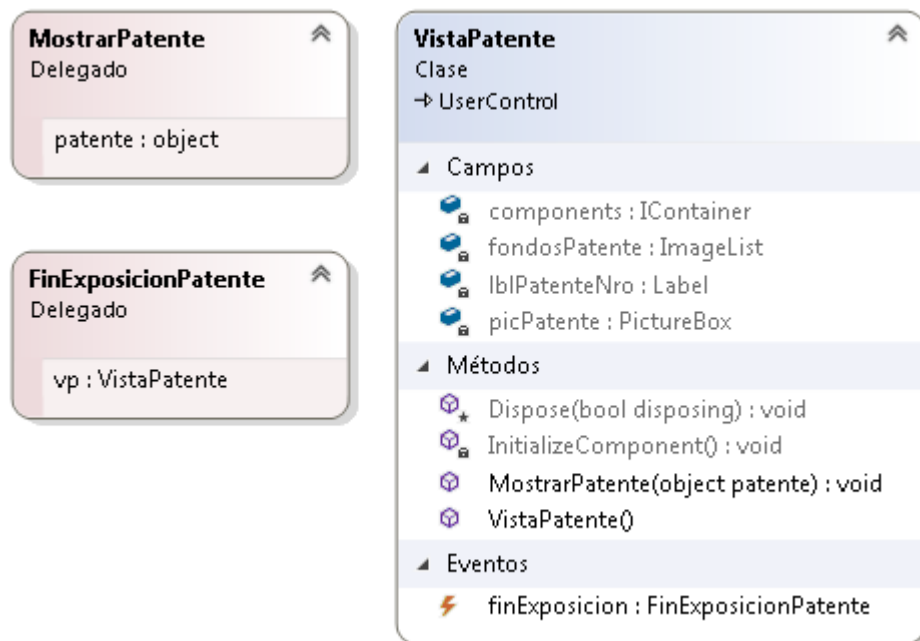
Entidades

- Dentro del proyecto **Entidades** se deberá respetar el siguiente esquema:



- Dentro de la clase PatenteStringExtension se deberá colocar un método de extensión llamado ValidarPatente.
 - ValidarPatente recibirá un string, y validará que coincida con los formatos de patente_mercosur o patente_vieja.
 - Si es patente_mercosur se deberá generar una nueva patente con tipo Mercosur.

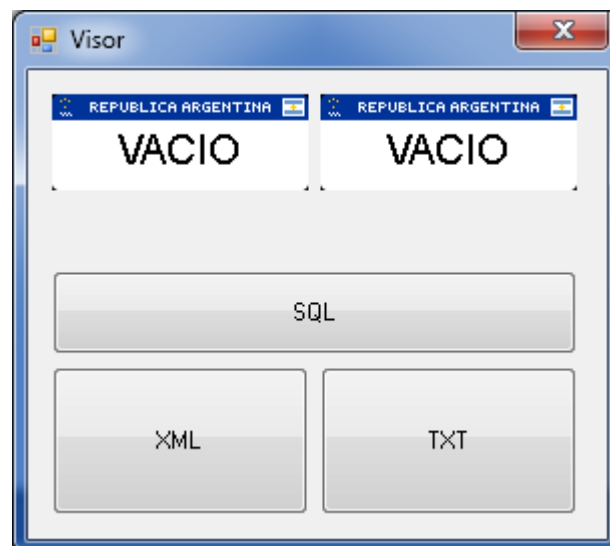
- c. Si es patente_vieja se deberá generar una nueva patente con tipo Vieja.
 - d. Caso contrario se lanzará una excepción del tipo `PatenteInvalidaException` con el mensaje "{0} no cumple el formato."
8. Dentro del proyecto VistaPatentes agregar al control de usuario la siguiente funcionalidad:



9. Agregar dos delegados al NameSpace (**por fuera de la clase**) con el siguiente formato `void FinExposicionPatente(VistaPatente vp)` y `void MostrarPatente(object patente)`.
10. Dentro del método `MostrarPatente` se deberá lograr que se muestre la patente durante un tiempo X y luego notificar por medio de un evento que finalizó dicha exposición. Completar debajo de los comentarios a fin de lograr la funcionalidad deseada.

Formulario

11. El formulario contará con 3 botones y dos controles de usuario. Al presionar los botones, se leerá de distintos orígenes una colección de patentes y se mostrarán en los controles de usuario:




12. Declarar un atributo del tipo lista de `Threads`. Inicializarlo en el constructor.
13. En el evento `Closing` del formulario, asegurarse de que todos los hilos estén terminados. Agregar el método `FinalizarSimulacion` que cumpla esa función.
14. Agregar el manejador del evento de la clase `VistaPatente` con el nombre `ProximaPatente`.
15. En el evento `Load` del formulario, asociar el evento `VistaPatente` para los objetos `vistaPatente1` y `vistaPatente2` con el manejador.

16. En cada botón leer del origen que corresponda (SQL, XML, TXT) agregar los datos a la cola de Patentes. También capturar las excepciones y llamar al método IniciarSimulacion.
17. IniciarSimulacion:
 - a. Finalizará los hilos activos.
 - b. Llamará al método ProximaPatente para cada uno de los objetos del tipo VistaPatente del formulario.
18. ProximaPatente, si hay elementos en la cola de patentes:
 - a. Instanciará un hilo parametrizado para el método MostrarPatente del objeto VistaPatente recibido.
 - b. Inicializará el hilo recién creado con el próximo elemento de la cola.
 - c. Agregará el hilo a la lista.

Test Unitarios

19. Agregar 3 test unitarios, uno para cada método Guardar de las clases Xml, Sql y Texto. Cada uno de estos test generará un nuevo archivo, luego lo leerá y verificará que el dato leído sea correcto.

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.