

**Universidad Tecnológica Nacional
Facultad Regional Avellaneda**



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	08/07/2021
Nombre:		Docente ⁽²⁾ :	
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<input type="checkbox"/> PP <input type="checkbox"/> RPP <input checked="" 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.

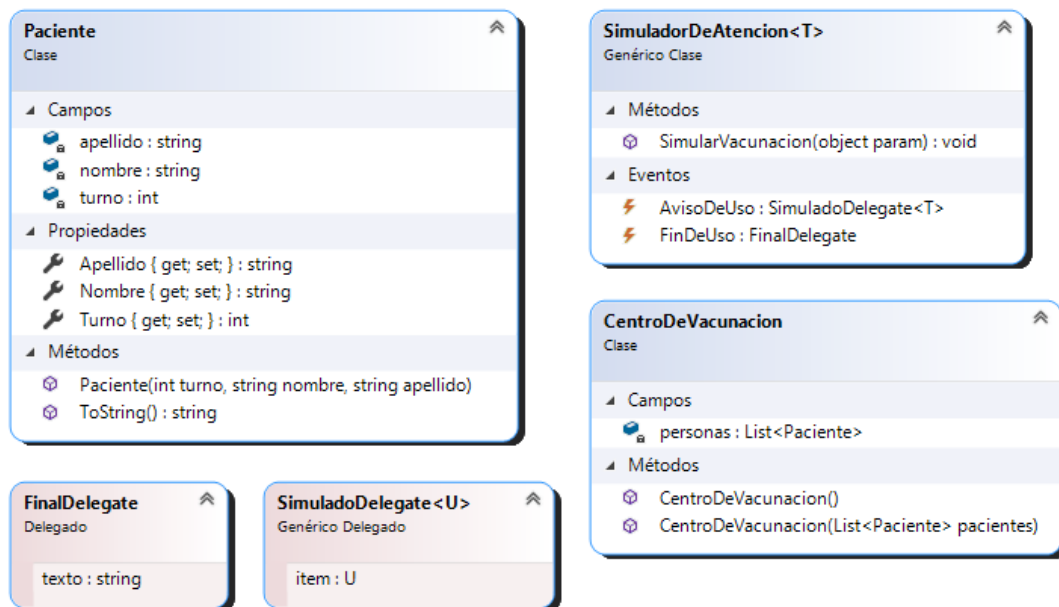
IMPORTANTE:

- **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, a no ser que se indique explícitamente otra cosa.
- 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 como se indicó anteriormente en este documento.
2. No agregar ninguna clase que no sea explícitamente requerida.

Esquema de Clases – Proyecto Entidades



SimuladorDeAtencion

3. Será la clase encargada de simular la atención. Para esto, el método SimularVacunacion será ejecutado en un hilo secundario y deberá:
 - a. Chequear el tipo recibido como parámetro, validando que sea del mismo tipo que su genérico.
 - b. Recorrer con foreach al argumento, previo casto `List<T> lista = (List<T>)param;`
 - i. Si hay suscriptores al evento AvisoDeUso, invocarlo.
 - ii. Hacer un Sleep de 1.2 segundos.
 - c. Al finalizar el foreach, ejecutar el evento FinDeUso con el mensaje "**Tarea finalizada**", siempre y cuando haya suscriptores.
4. Al finalizar guardará en un archivo de texto la información de los pacientes atendidos.

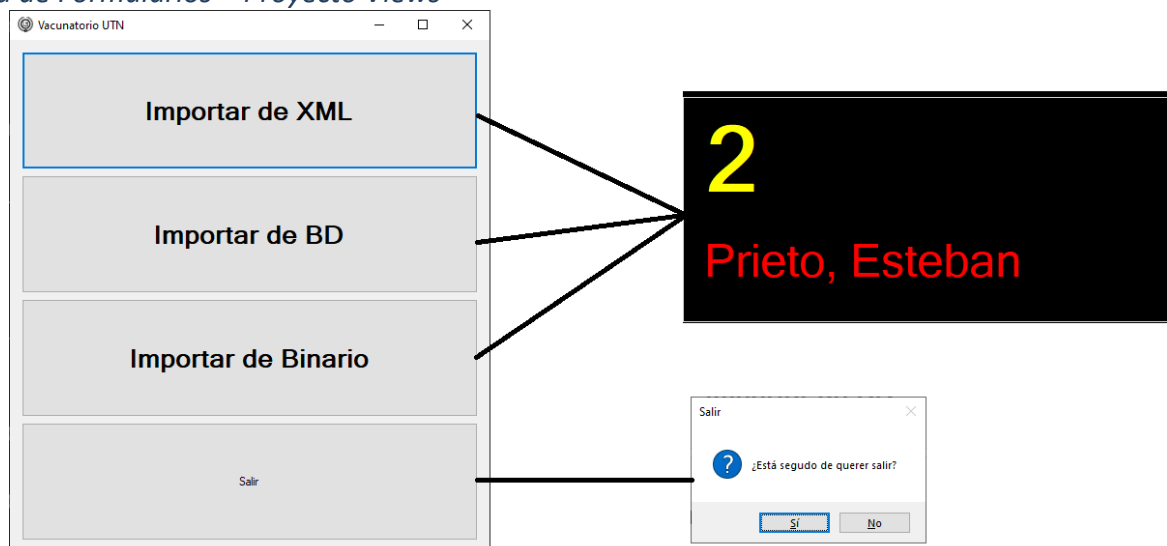
Paciente

1. ToString retornará "Apellido, Nombre".

General

2. Agregar lo necesario para la serialización que se pide más adelante.
3. Crear una interfaz original, por ejemplo para el manejo de archivos y base de datos.
4. Ubicar los métodos donde creo correcto.

Esquema de Formularios – Proyecto Views



5. La visualización debe ser exactamente esta.
6. En FrmLlamador figurará arriba y en amarillo el número de Turno y debajo Apellido, Nombre.
7. FrmLlamador implementará los métodos necesarios que permita simular el llamado por pantalla de pacientes obtenidos desde la BD, XML o BIN. El hilo es parte de la lógica de FrmLlamador. Este formulario también implementará los manejadores para los eventos de SimuladorDeAtencion.
8. Cada botón importar lo hará desde el medio especificado.
 - a. Para Binario y XML deberá utilizar OpenFileDialog.
 - b. Solo se abrirán archivos válidos. Caso contrario se lanzará la excepción propia ArchivoInvalidoException (crearla dentro del proyecto Entidades).
 - c. Los archivos contendrán una lista de Pacientes.
 - d. Para BD utilizar la siguiente base de datos SQL:


```
USE [master]
GO
CREATE DATABASE [20210708-SP]
GO
USE [20210708-SP]
GO
CREATE TABLE [dbo].[Pacientes](
    [turno] [int] NOT NULL,
    [nombre] [varchar](50) NOT NULL,
    [apellido] [varchar](50) NOT NULL
) ON [PRIMARY]
GO

INSERT INTO Pacientes(turno, nombre, apellido) values
(7,'Guillermo','Barros Schelotto'),
(8,'Diego','Cagna'),
(9,'Martín','Palermo'),
(10,'JR','Riquelme')
```
 - e. Al finalizar la importación de cada medio (binario, xml o bd), se ejecutará FrmLlamador.
9. El botón salir o la cruz pedirán validación (Está seguro? Si/No) en la salida.

Test Unitarios

10. Realizar un test unitario que pruebe la excepción ArchivoInvalidoException.
11. Realizar un test unitario que pruebe la lectura de algún archivo (binario o XML).
12. Realizar un test unitario libre, comprobando alguna funcionalidad útil del sistema. Se juzgará lo útil del método.

Al finalizar, colocar la carpeta de la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre Apellido.Nombre.division.zip y compartir este por Slack sólo con el docente titular de la cursada.