

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	23-06-2022
Nombre:		Docente ⁽²⁾ :	
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP SP RSP FIN </div>		

(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.
- El proyecto debe ser creado en .Net 5.
- Colocar sus datos personales en el nombre de la carpeta principal y la solución: Apellido.Nombre.Div. Ej: Pérez.Juan.2D. No se corregirán proyectos que no sea identificable su autor.
- 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.
- Aplicar los principios de los 4 pilares de la POO.
- Cuando aparezcan ejemplos de textos por pantalla, figurarán entre {} los datos a reemplazar por variables.

Tomando como referencia el que fuera nuestro primer parcial, del cual se adjunta el proyecto para utilizar como base:

Atención de pacientes
×

Personal médico

Dávila, Fede
Prieto, Esteban

Pacientes

Cagna, Diego
Moreno, Alfredo
Giunta, Blas
Trelles, John Jairo

Atender

Prieto, Esteban
EDAD:70
¿Finalizó residencia?NO
ATENCIONES:
Cagna, Diego
EDAD:23
Recide en: Tigre
Paciente curado

Salir

Atención finalizada

17/03/2022 15:51:16 se ha atendido a Giunta, Blas

Aceptar

- Leer los médicos y pacientes de la base de datos provista.
- Cambiar la forma de cargar la lista de pacientes:
 - Utilizar Task, Eventos y Delegados.
 - Crear una clase Mock e instanciarla en el formulario.
 - Utilizar Random.
 - Cada tiempo random (entre 1 y 5 segundos) agregar un paciente de los leídos de la BD al ListBox.
 - Al atender a dicho paciente, este será borrado del ListBox.
- Agregar un menú superior que muestre la opción:
 - Archivo:
 - Exportar: exportará en XML al médico seleccionado, con todas sus atenciones, guardando el archivo con como "nombre-alumno.xml" (ej: juan-perez. xml) en el escritorio de la máquina.
- Crear una interfaz genérica que se utilizará en la clase de lectura/escritura de xml y en la clase que maneje la base de datos.
 - Si hay métodos no implementados, lanzar la excepción correspondiente.
- Recordar manejar las excepciones donde corresponda.
- Crear un Test Unitario para probar el archivo exportado en XML. Para esto deberá:
 - Crear un médico ficticio y al menos 2 atenciones (no hace falta leer nada de la base de datos).
 - Exportar a XML.
 - Leer el archivo XML y validar que los datos sean correctos. Mínimamente comprobar nombre y apellido del médico, y cantidad de atenciones.
- Crear un Test Unitario que valide la conexión a la base de datos, sin repetir el string connection.
- Crear un Test Unitario que pruebe un método no implementado de la base de datos.
- Punto extra: agregar como método de extensión de nuestra clase de manejo de archivos:
 - Método BorrarMedico que elimine un archivo dado.
 - Agregar su uso al Test Unitario que comprueba el guardado y lectura del archivo xml para que elimine este archivo de prueba al finalizar.

Al finalizar, colocar 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.