

Universidad Tecnológica Nacional
Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	18-06-2020
Nombre:		Docente ⁽²⁾ :	Dávila-Cerizza-Rodríguez
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> <div>PP</div> <div></div> <div>RPP</div> <div></div> <div>SP</div> <div>X</div> <div>RSP</div> <div></div> <div>FIN</div> <div></div> </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.
- 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.

1) Requerimiento:

Nuestro cliente, una clínica privada de la Ciudad Autónoma de Bs As nos ha solicitado desarrollar un sistema que permita llevar adelante el control.

El Sistema debe tener la capacidad de:

- Asignar un nuevo turno a la cola de turnos.
- Mostrar en tiempo real que paciente está siendo atendido.
- Mostrar próximo paciente a ser llamado.
- Guardar un archivo de texto los errores que vayan surgiendo (log).
- Grabar en la base de datos la información de la consulta una vez que esté terminada.

El cliente pide que el programa conste de tres formularios que se muestren en pantalla al mismo tiempo (no quiere tres proyectos diferentes, quiere que el mismo Proyecto, al iniciar, muestre tres formularios diferentes) en los cuales podrá obtener las funcionalidades requeridas anteriormente.

2) Análisis funcional:

- La clínica cuenta con una base de datos llamada ClinicaUTN con la siguiente información:
 - Tabla Especialistas: Contiene información de los especialistas.
 - Posibles especialidades:
 - Oftalmologia,
 - Odontologia,
 - Otorrinolaringologia,
 - Ginecologia,
 - Dermatologia,
 - Obtetricia,
 - Pediatria,
 - Psicologia,
 - ClinicaGeneral
 - Tabla Pacientes: Información de pacientes que existan en sistema previamente.
 - Posibles estados de obra social:
 - NoTiene,
 - CoberturaBasica,
 - CoberturaCompleta
 - Tabla Turnos: Guardará registros de las consultas que fueron realizadas.

Nota: La base de datos le será proporcionada en conjunto con este documento para que pueda replicarla localmente en una base de datos de prueba.

- Debido a remodelaciones de otra sucursal, la clínica deberá (al momento de iniciar el programa) detectar pacientes y especialistas que hayan sido derivados desde aquella sucursal a esta, levantando sus datos por medio de una deserialización de un archivo XML que le será adjunto justo a este documento.
Nota: La aplicación deberá poder levantar siempre los archivos del siguiente path:
"MisDocumentos/SegundoParcialUtn/ArchivosClinica/".
- Con la finalidad de poder probar el programa, se simulará que los turnos duran 10 segundos.

Pasados estos 10 segundos deberá insertarse en la base de datos Turnos el resultado de la consulta SI el paciente existe en la base de datos de la clínica.

En caso de que el paciente NO pertenezca a la clínica (es decir, haya sido levantado desde el archivo XML adjunto en este documento) deberá serializarse el turno en un archivo de tipo XML con fecha - apellido – nombre del paciente (path:

“MisDocumentos/SegundoParcialUtn/TurnosClinica/”).

Luego de esto, el especialista llamará al paciente que estaba próximo a ser llamado, y deberá actualizarse la información del formulario que mostrará nombre y apellido del próximo paciente a ser llamado.

- La aplicación deberá de tener implementado Unit Testing para poder probar como mínimo la siguiente funcionalidad:
 - Serialización y deserialización de turnos en XML y Binario.
 - Alta de nuevos turnos
- Aclaraciones:
 - Queda a criterio del alumno la forma en la que desee gestionar la carga de los turnos para probar el sistema. Puede ser de forma manual o bien tener turnos hardcodeados.
 - Al momento de serializar o de escribir el archivo (es decir, cuando creamos files), deberán verificar si la ruta existe. De no existir, crearla la ruta, grabar el file y al finalizar, lanzar una excepción personalizada que grabe en el .txt de logs, la creación de la ruta por no existir.

Criterios de evaluación mínimos

Se tendrán en cuenta:

- La solución deberá compilar sin errores.
- No puede haber funcionalidad que rompa el programa.
- Uso de validaciones.
- Uso conceptos de POO vistos hasta el momento en clase.
- Uso de las reglas de estilo vistas en clase.
- Deberán utilizarse las buenas prácticas de programación explicadas y practicadas durante la cursada.
- Deberá estar comentado el código (Summary).
- Los formularios NO pueden tener color por defecto.
- Deberán implementarse además de los conceptos evaluados en el primer parcial, los vistos durante la segunda parte del cuatrimestre:
 - Manejo de excepciones
 - Serialización de objetos
 - Escritura de archivos
 - Generics
 - Unit Testing
 - Interfaces
 - Threads
 - Eventos
 - Delegados
 - Métodos de extensión
- Los nombres de los formularios y sus títulos NO pueden ser el por defecto (Los controles NO pueden tener nombre por defecto y DEBEN ser descriptivos).

Aquellos parciales que estén en condiciones de aprobar luego de ser revisados por el docente asignado, serán llamados a una etapa de exposición donde mostraran su código, se les harán preguntas sobre el código, sobre el funcionamiento del sistema, y se les hará la devolución pertinente.

A partir de este examen, se solicita que la persona evaluada tenga funcionando la cámara web durante el meet.