

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	11-05-2022
Nombre:		Docente ⁽²⁾ :	
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP X RPP </div>	<div style="display: flex; justify-content: space-around;"> 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.

Vamos a crear un sistema de atención a pacientes simplificada, para lograr una funcionalidad determinada en el tiempo disponible. Para esto, en principio, propondremos tomar algunos atajos a fin de no perder tiempo en cosas de baja importancia:

Atención de pacientes
×

Personal médico

Dávila, Fede

Prieto, Esteban

Pacientes

Cagna, Diego

Moreno, Alfredo

Giunta, Blas

Trelles, John Jairo

Atender

Salir

Prieto, Esteban

EDAD:70

¿Finalizó residencia?NO

ATENCIONES:

Cagna, Diego

EDAD:23

Recide en: Tigre

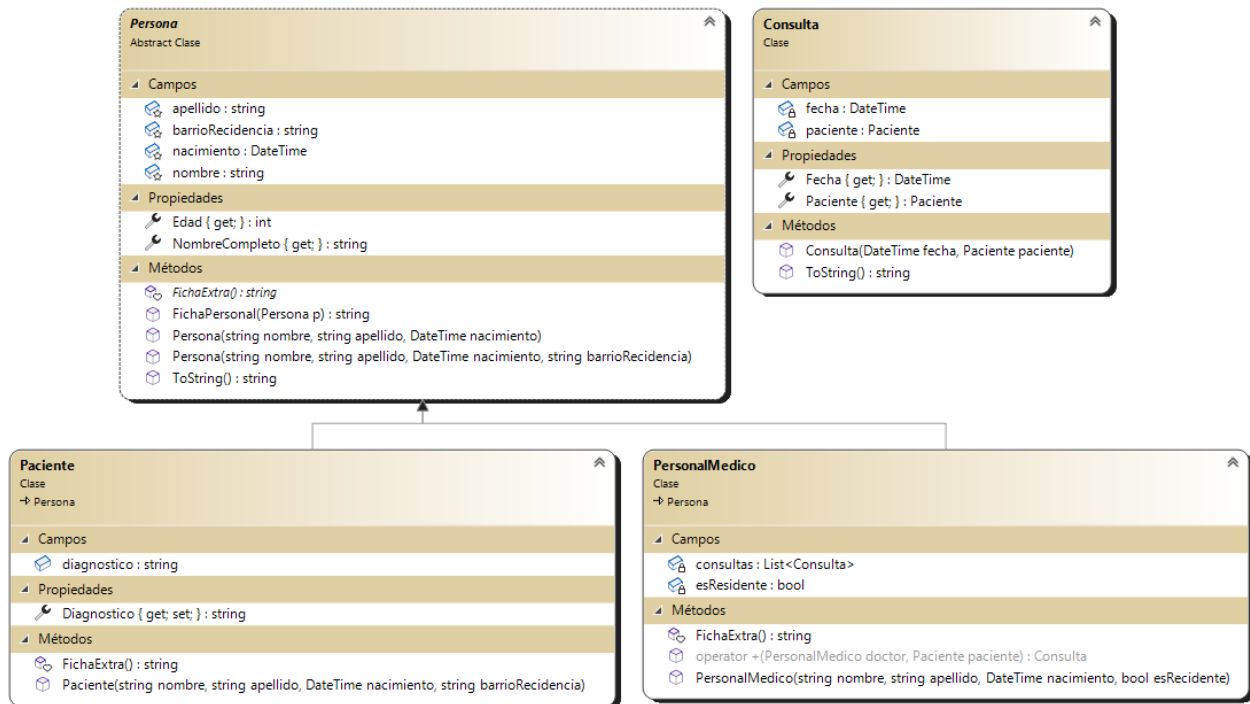
Paciente curado

Aceptar

Atención finalizada

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

1. Para comenzar, crear un proyecto del tipo Biblioteca de Clases y reproducir el siguiente esquema:



Persona

2. Propiedades:

- NombreCompleto:** de solo lectura. Retornará el nombre completo de la persona con el formato Apellido, Nombre. Utilizar String Format.
- Edad:** de solo lectura. Retornará el resultado de `return DateTime.Today.AddTicks(-this.nacimiento.Ticks).Year - 1`.

3. Métodos:

- ToString:** retornará el nombre completo con el formato Apellido, Nombre.
- FichaPersonal:** debe armar una ficha con todos los datos del objeto instanciado con el formato:
 {Apellido, Nombre}
 EDAD: {Edad}
 {Datos del objeto instanciado}
- FichaExtra:** internal y abstracto.

PersonalMedico

4. **FichaExtra:** debe armar una ficha con todos los datos del objeto con el formato:

¿Finalizó residencia? {SI/NO}

ATENCIONES:

{Fichas de la lista de CONSULTAS}

5. El operador + agregará nuevas consultas a la lista, tomando como fecha de inicio de la misma el método Now de la clase DateTime.

Paciente

6. **FichaExtra:** debe armar una ficha con todos los datos del objeto con el formato:

Recide en: {barrio}

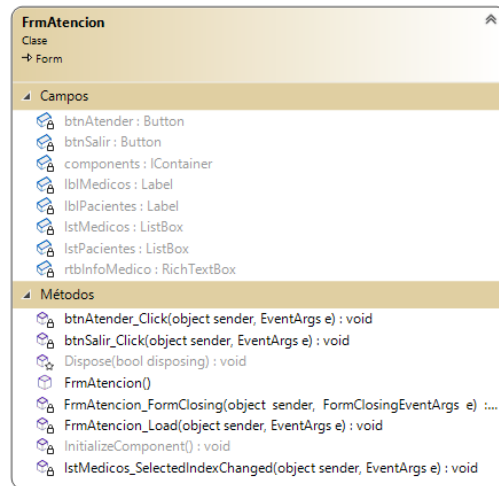
{diagnóstico}

Consulta

7. ToString: {Fecha} se ha atendido a {Nombre completo del paciente}

Formulario

8. Crear, en un proyecto separado, el formulario visto en la primer hoja de este documento.
9. El formulario deberá tener en el título el nombre del alumno.
10. El formulario deberá iniciar centrado en la pantalla, teniendo ancho suficiente como para que se lean los textos.
11. Respetar título y diseño, así como los nombres de sus atributos como muestra el siguiente diagrama de funcionamiento mínimo requerido:



12. El método controlador del Load del formulario deberá tener el siguiente código:

```
lstMedicos.Items.Add(new PersonalMedico("Fede", "Dávila", new DateTime(1999, 12, 12), false));  
lstMedicos.Items.Add(new PersonalMedico("Esteban", "Prieto", new DateTime(1951, 11, 12), true));  
lstPacientes.Items.Add(new Paciente("Diego", "Cagna", new DateTime(1998, 6, 16), "Tigre"));  
lstPacientes.Items.Add(new Paciente("Alfredo", "Moreno", new DateTime(1989, 1, 21), "DF"));  
lstPacientes.Items.Add(new Paciente("Blas", "Giunta", new DateTime(1912, 12, 12), "La Boca"));  
lstPacientes.Items.Add(new Paciente("John Jairo", "Trelles", new DateTime(1978, 8, 30), "Medellín"));
```

13. El botón atender deberá:

- a. Validar que se hayan seleccionado un médico y un paciente. Caso contrario mostrar el mensaje "Debe seleccionar un Medico y un Paciente para poder continuar." con el título "Error en los datos". Mostrar solo botón de aceptar e ícono de error.
- b. Una vez validado, generar la consulta:
 - i. El diagnóstico será "Paciente curado" para todos los casos.
 - ii. Se deberán deseleccionar ambos ListBox.
 - iii. Mostrar un mensaje que muestre la información de la consulta, su título sea "Atención finalizada" y solo el botón de aceptar.

14. Al seleccionar un médico, se deberá mostrar su ficha en el RichTextBox ubicado en la zona baja del formulario.

15. Al intentar salir por cualquier medio, se deberá preguntar "¿Seguro que desea salir?", con título "Salir", opciones por si o no, e ícono de pregunta. De más está decir que si contesta que NO se deberá permanecer con el formulario abierto.

PUNTO EXTRA:

- Modificar diagnóstico para la clase Paciente generando un enumerado con los valores
 - Curado

- Gripe
 - Coronavirus
- Asignar el diagnóstico por intermedio de un Random.
- Hacer las modificaciones para que esto quede funcionando.

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.