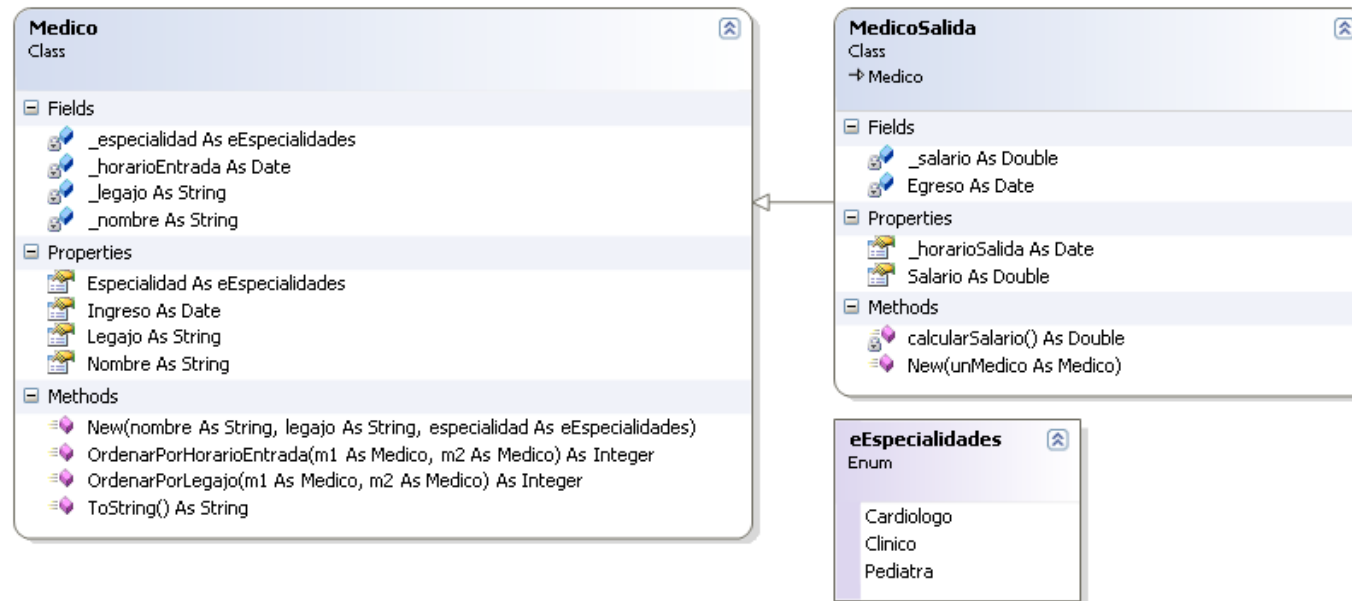


Recuperatorio Laboratorio III 2016 (Primer Parcial)

ADMISTRACIÓN DE MEDICOS

Esta aplicación nos permitirá controlar el ingreso y egreso de los médicos que trabajan en un hospital. Además se podrá ordenar el listado, modificar datos de los médicos y eliminarlos del listado.

Realizar en un proyecto de tipo 'Class Library' las clases **Medico**, **MedicoSalida** y el enumerado **eEspecialidad**, respetando los diagramas que se muestran a continuación:



Nota Técnica N°1: La propiedad Salario será de solo lectura y llamara al método calcularSalario. Este método utilizara el método de instancia de Datetime Subtract para obtener el intervalo de tiempo TimeSpan entre el ingreso y egreso del médico para calcular los minutos trabajados, y en base a ese valor se aplicarán los siguientes montos:

Especialidad	Valor x min.
Cardiólogo	\$55.00
Clínico	\$50.00
Pediatra	\$45.00

Añadir a la misma solución un proyecto de tipo **‘Windows Forms’**, donde se crearán: un módulo de inicio y **5 (cinco)** formularios.

(FrmPrincipal, FrmGestion, FrmMostrar, FrmMedico y FrmMedicoHeredado)

FRMGESTION

The screenshot shows the 'FrmGestion' window with a single list box labeled 'lstMedicos'.

Este formulario tendrá una lista genérica de tipo **Medico** y otra de tipo **MedicoSalida** como atributos **protegidos**. El combo se cargará con valores literales y no se podrá escribir en él, y el elemento seleccionado por defecto será **“Legajo”**.

FRMPRINCIPAL:

The screenshot shows the 'Principal' window with a menu bar containing 'Gestion Diaria', 'Mostrar', and 'Salir'.

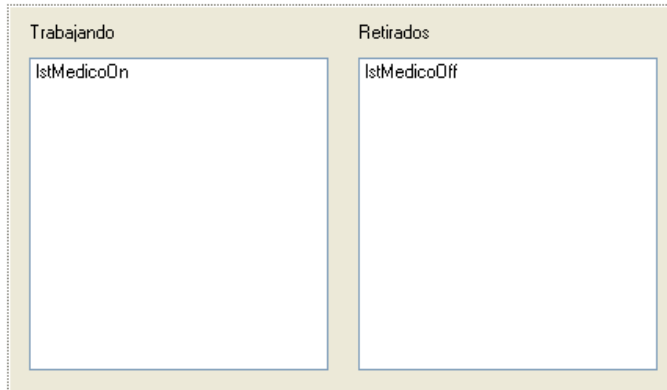
EL FORMULARIO FRMMEDICO Y EL QUE HEREDA DE ESTE:

The screenshot shows two overlapping windows: 'FrmMedico' and 'FrmMedicoHeredado'.
FrmMedico fields: Legajo (text), Nombre (text), Especialidad (dropdown), Aceptar, Cancelar.
FrmMedicoHeredado fields: Legajo (text, value: 123), Nombre (text, value: Juan Perez), Especialidad (dropdown, value: Pediatra), Pago (text, value: 480), Aceptar, Cancelar.

El formulario FrmMedico tendrá una propiedad de **sólo lectura** de tipo **“Medico”**, y el combo se cargará en un método, recorriendo el enumerado. En este combo no se permitirá escribir y el elemento que aparecerá por defecto será **“Pediatra”**. El formulario heredado tendrá una propiedad de **sólo lectura** de tipo **“MedicoSalida”**.

FRMMOSTRAR:

Este formulario tendrá como atributos **privados** dos listas genéricas de tipo **Medico** (_medicos) y de tipo **MedicoSalida** (_salidas) con sus respectivas propiedades de **sólo escritura**.



Nota Tecnica Nº2: Todos los controles de los formularios serán “Protegidos” (Protected)

- 1) Formulario Principal:
 - a) Al seleccionar la opción de FrmPrincipal **‘Gestion Diaria’**, se mostrará una instancia de FrmGestion (**de forma modal**).
 - b) Al pulsar la opción **‘Mostrar’** se deberá mostrar una instancia de FrmMostrar (para que **FrmPrincipal sea su dueño**).
 - c) Al presionar el botón para cerrar **FrmPrincipal** (la “x”), consultaremos si realmente desea salir de la aplicación, si se contesta que **“NO”**, cancelaremos el **cerrado de la aplicación. La opción Salir del menú tendrá el mismo efecto. (No repetir código)**
- 2) Al hacer click en el botón “Ingreso” del formulario **FrmGestion**, se instanciará **FrmMedico** y se lo mostrará de forma modal, y si es **OK**, se agregará un medico a la lista genérica de tipo **Medico** de **FrmGestion**.
- 3) Al seleccionar un criterio de ordenamiento, se creará un **delegado** de tipo **“Comparison (Of Medico)”** con la dirección del método que corresponda y se procederá a ordenar la lista y luego mostrarla.
- 4) Los manejadores del **ListBox** y los botones “Modificación” y “Egreso” **No deben tener manejadores agregados de forma estática**. Agregar y quitar manejadores de **forma dinámica** de tal forma que **NO generen errores**: en tiempo de ejecución, de índices al hacer click en los botones “Modificacion” y el “Egreso”, como así también en el ListBox.
- 5) En el botón “Modificación” mostraremos una instancia del formulario **FrmMedico** con todos los datos del médico seleccionado en el ListBox para ser modificado. Recordar que la propiedad **Medico** del formulario debe ser de **sólo lectura** y sus controles **“Protegidos”**

- 6) En el botón “Egreso” se mostrará una instancia del formulario heredado (***FrmMedicoHeredado***) con el médico seleccionado en el ListBox y se procederá al cálculo del salario a través de la propiedad “Salario”. Si se pulsa **Aceptar**, se creará un objeto de tipo **MedicoSalida** y se lo expondrá en la propiedad de **sólo lectura** del formulario, para ser agregado a la lista genérica de tipo **MedicoSalida**. Modificar el manejador del evento click del botón **Aceptar** de FrmMedico para declararlo como **‘Virtual’**. Al mismo tiempo, se debe quitar al médico de la lista genérica de tipo **Medico**. Además toda la información se guardará en un archivo de texto que se ubicará en el mismo directorio donde se está ejecutando la aplicación.
- 7) La instancia de **FrmMostar** deberá tener la propiedad **FormBorderStyle=“NONE”**. Este formulario tendrá un sólo método público llamado **“ActualizarListados”** que recibirá una lista genérica de tipo **Medico** y otra de tipo **MedicoSalida** como parámetros. Al invocar este método se actualizarán los ListBoxes (**lstMedicoOn** con los nombres y el horario de ingreso de los médicos de la lista genérica de tipo Medico y **LstMedicosOff** con los legajos, Horario de ingreso, egreso y salario de los medicos de la lista genérica de tipo MedicoSalida).
- a) Crear un tipo de **delegado** para poder invocar al método **“ActualizarListados”** en un módulo público.
 - b) La variable de instancia de este tipo de delegado será creada en el mismo módulo.
 - c) La variable se instanciará en FrmPrincipal, en la opción de menú ‘Mostrar’, apuntando al método de la instancia de FrmMostrar.
 - d) Invocar al delegado al Ingresar, modificar o egresar un elemento de la lista de medicos.
- 8) Crear la clase “ComboEspecialidades”, que herede de la clase “ComboBox”, que en su constructor cargue las especialidades del enumerado Entidades.eEspecialidades.

Esta clase se construirá en una solución llamada “MiControl” y se creará en un proyecto de tipo Librería de Clases.
Reemplazar el control ComboBox (cboEspecialidad) del formulario “frmMedico” por este nuevo control.

La Solución se deberá guardar en una carpeta nombrada con su Apellido y nombre. Se deberá incluir un método Main, el cual tendrá un bloque Try-Catch-Finally, y por el cual se iniciará la aplicación.

Tiempo máximo para resolver el examen: 90 minutos.