

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

Materia: Laboratorio de Programación II

Apellido:		Fecha:	25/06/2019
Nombre:		Docente ⁽²⁾ :	F. Dávila / D. Boullon
División:	2ºD	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	<div style="display: flex; justify-content: space-around;"> PP RPP SP X RSP FIN </div>		


(1) Las instancias válidas son: 1º Parcial (PP), Recuperatorio 1º Parcial (RPP), 2º Parcial (SP), Recuperatorio 2º Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

IMPORTANTE:

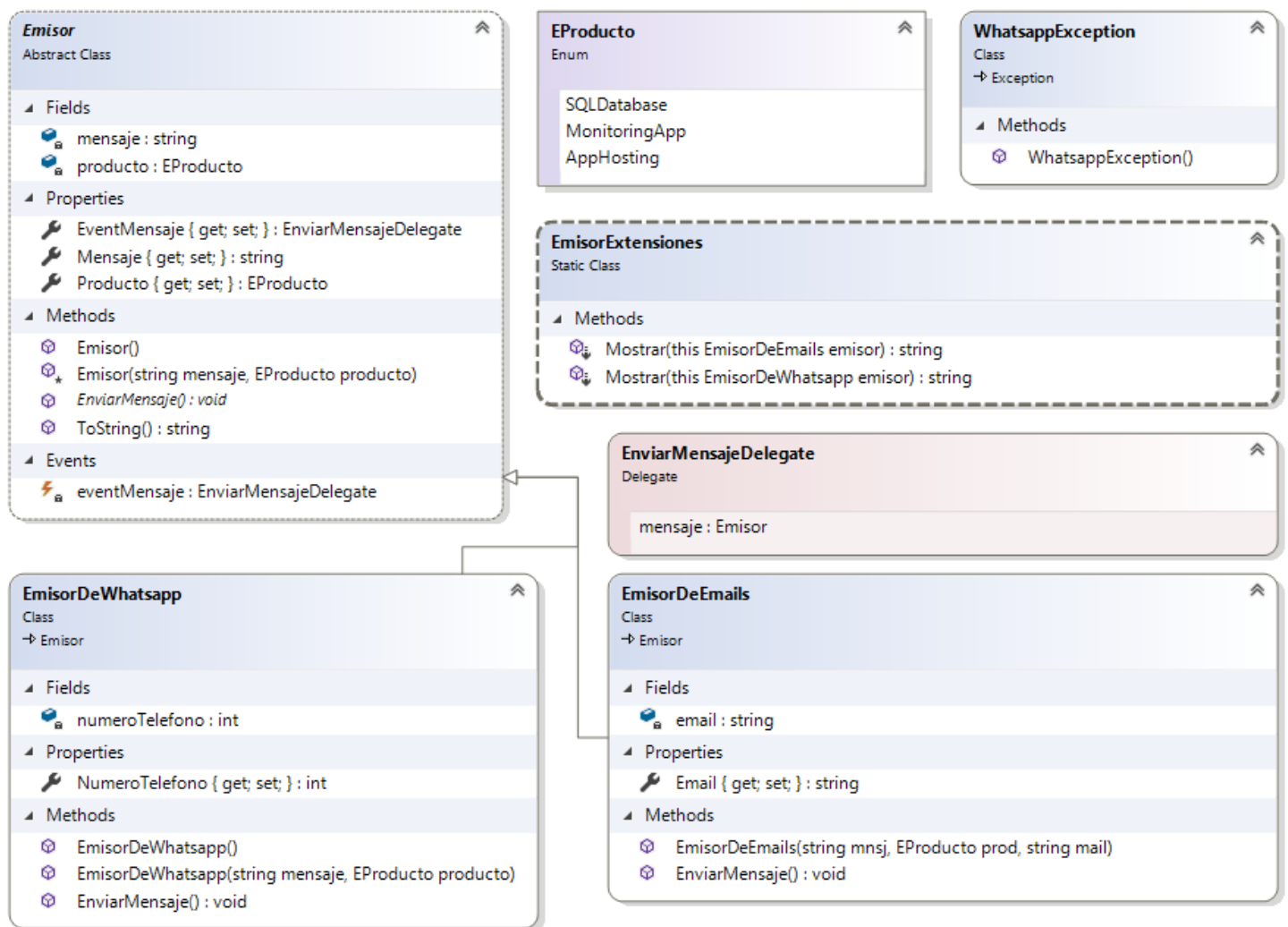
- Guardar el proyecto en el **disco D:**. Ante un corte de energía o problema con el archivo de corrección, el alumno será responsable de que el proyecto sea recuperable.
- **2 (dos) errores en el mismo tema anulan su puntaje.**
- **Errores de conceptos de POO anulan el punto.**
- **Cada tema vale 1 (un) punto (Herencia, Generics, Test Unitarios, etc.). La correcta documentación también será evaluada.**
- **Se deberán tener al menos el 60% bien de los temas a evaluar según la instancia para lograr la aprobación.**
- Colocar sus datos personales en el nombre del proyecto principal, colocando: Apellido.Nombre.AñoCursada. Ej: Pérez.Juan.2018. No se corregirán proyectos que no sea identificable su autor.
- **Salvo que se indique lo contrario, TODAS** las clases deberán ir en una Biblioteca de Clases llamada Entidades.
- No se corregirán exámenes que no compilen.
- **Reutilizar** tanto código como crean necesario.

Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP que deberá tener como nombre

Apellido.Nombre.AñoCursada.zip y dejar este último en el Escritorio de la máquina. Luego presionar el botón  de la barra superior, **colocar un mensaje** y presionar *Aceptar*. **Aguardar a que el profesor indique que el examen fue copiado de forma correcta.** Luego retirarse del aula.

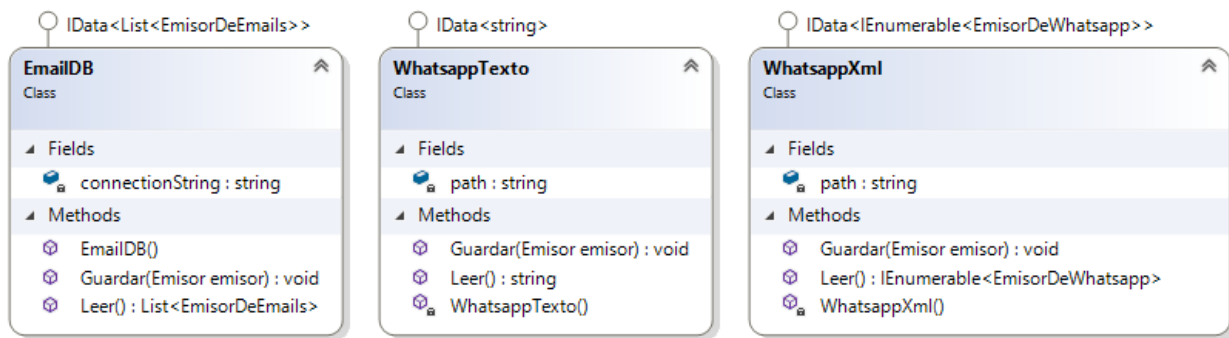
TIEMPO MÁXIMO PARA RESOLVER EL EXAMEN 90 MINUTOS.

1. Editar el nombre de la solución proporcionada Solución llamada "20190625.[Apellido].[Nombre]" siendo el Apellido y Nombre del alumno que lo crea.
2. Debe haber 4 proyectos: Biblioteca de clases **SP.Data**, *Biblioteca de Clases* **SP.Entidades**, *Formulario* **SP.Form** y test unitario **SP.Test**.
3. Agregar elementos faltantes para obtener el siguiente diagrama de clases en SP.Entidades:



4. Crear un delegado llamado EnviarMensajeDelegate que retorna void y recibe un Emisor.
5. Emisor: completar con:
 - a. eventMensaje: evento del tipo EnviarMensajeDelegate.
 - i. Agregar la etiqueta [field: NonSerialized] sobre el event para evitar que se intente serializar.
 - b. Agregar la property (get y set) para el evento eventMensaje.
 - i. Agregar la etiqueta [XmlIgnore] sobre la propiedad
 - c. Esta clase debe poder ser serializada.
6. EmisorDeEmails: hereda de **Emisor** y tiene 1 atributo privado.
 - a. EnviarMensaje(): valida si hay manejadores, en caso de que no haya, lanza un EmisorException indicando que no hay manejadores. Caso contrario duerme el Thread 3 segundos y luego invoca al evento EventMensaje pasándose a sí mismo (*Invoke(this)*) como parámetro.
7. EmisorDeWhatsapp:
 - a. La propiedad **NumeroTelefono** permite cargar un número de teléfono siempre que esté entre 1500000000 y 1599999999, si el valor ingresado esta por fuera del rango lanza un WhatsappException indicando que el numero esta por fuera del rango.
 - b. **EnviarMensaje**: duerme el hilo durante 1 segundo y luego invoca al evento EventMensaje pasándose a sí mismo (this) como parámetro. En caso de que no esté cargado el numero lanza un WhatsappException indicando que el número no fue cargado.
 - c. Debe ser serializable.
8. EmisorExtensiones:
 - a. Tiene dos métodos llamados Mostrar(...) que extienden a EmisorDeWhatsapp y EmisorDeEmail. Estos métodos retornarán la información devuelta por ToString() de la clase padre y sumará el teléfono o email ("- {telefono/email}") según corresponda.

En el proyecto SP.Data generar las siguientes entidades:



9. **IData<T>**: Interface genérica con los siguientes métodos:

- Guardar(Emisor emisor), retorna void.
- Leer<T>()

10. **EmailDb** (Implementa **IData<T>**):

- Tiene un constructor de instancia que setea el valor de coneccion al connectionString.
- Guardar (), recibe un emisor lo castea a EmisorDeEmail y lo guarda en la base de datos.
- Leer (), recupera la información de la base de datos y devuelve una lista de EmisoresDeEmails

11. **WhatsappTexto** (Implementa **IData<T>**):

- Constructor de instancia que le asigna al atributo "path" la ruta de una carpeta llamada whatsappTexto.txt en el escritorio (eventualmente se creara).
- Guardar(): agrega al archivo (si no existe lo crea) el resultado del método de extensión Mostrar().
- Leer(): retorna el contenido en como string.

12. **WhatsappXml** (Implementa **IData<T>**):

- Constructor de instancia que le asigna al atributo "path" la ruta de una carpeta llamada whatsapp.xml en el escritorio (eventualmente se creara).
- Guardar(): agrega al archivo (si no existe lo crea) el objeto serializado.
- Leer(): recupera el o los objetos serializados y los devuelve en forma de lista.

13. Crear la base de datos SP_2019 y ejecutar el siguiente escript:

```
USE [SP_2019]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Emails2](
    [mensaje] [varchar](250) NOT NULL,
    [producto] [varchar](50) NOT NULL,
    [email] [varchar](50) NOT NULL
)
GO
```

14. En el siguiente formulario desarrollar las funcionalidades...:

The screenshot shows a Windows Forms application window titled "Nombre.Apellido". The form contains several controls: a text box for "Email", a text box for "Tel.", a dropdown for "Producto", a large text area for "Mensaje", and four buttons: "Enviar Email", "Email Info", "Enviar Whatsapp", and "Whatsapp Info". Below these controls is a label "Output" and a large text area for displaying results.

To the right of the form is the class diagram for **FrmSender**. It lists the following fields: `btnSendEmail : Button`, `btnSendWhat : Button`, `btnShowMail : Button`, `btnShowWhat : Button`, `cmbProducto : ComboBox`, `components : IContainer`, `lblEmail : Label`, `lblMensaje : Label`, `lblOutput : Label`, `lblProducto : Label`, `lblTel : Label`, `richMensaje : RichTextBox`, `richOutPut : RichTextBox`, `threads : List<Thread>`, `txtEmail : TextBox`, and `txtTel : TextBox`. The methods listed are: `BtnEnviarEmail_Click(object sender, EventArgs e) : void`, `BtnEnviarWhatsapp_Click(object sender, EventArgs e) : void`, `BtnShowEmails_Click(object sender, EventArgs e) : void`, `BtnShowWhatsapps_Click(object sender, EventArgs e) : void`, `Dispose(bool disposing) : void`, `FrmSender()`, `InitializeComponent() : void`, and `MostrarMensaje(Emisor mensaje) : void`.

- a. Tiene un atributo privado `List<Thread> threads`, que coleccionara todos los threads que se generen.
 - b. Usar Try Catch para todos los escenarios en que puede lanzarse una excepcion.
 - c. **MostrarMensaje:** recibe un objeto del tipo Emisor y muestra en un MessageBox el siguiente mensaje: *`\${mensaje.GetType().Name} - Se envió mensaje del producto: {mensaje.Producto}`*
 - d. El botón **Enviar Email** debe crear un EmisorDeEmails, agregar al **eventMensaje** el manejador "EmailDB.Guardar" y "MostrarMensaje" y ejecutar el método **EnviarMensaje** (del objeto EmisorDeEmails) en un hilo nuevo. *(Este método invocará al evento.)*
 - e. El botón **Enviar Whatsapp** hace lo mismo con el objeto EmisorDeWhatsapp per agrega al event eventMensaje el manejador "WhatsappTexto.Guardar" y "**MostrarMensaje**" y ejecutar el método **EnviarMensaje** (del objeto EmisorDeEmails) en un hilo nuevo. *(Este método invocará al evento.)*
 - f. Email info y Whatsapp info utilizarán las clases EmailDb y WhatsappTexto para traer la información de la base de datos y el archivo de texto dependiendo el caso.
 - g. Cuando se cierra la aplicación abortar todos los hilos.
 - h. Catchear todas las posibles excepciones y mostrar los errores en un MessageBox.
15. Crear un proyecto y testear dos escenarios que puedan lanzar las excepciones.