

Universidad Tecnológica Nacional Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos


Materia: Laboratorio de Programación II

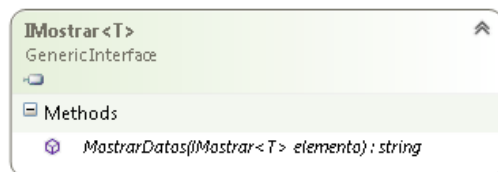
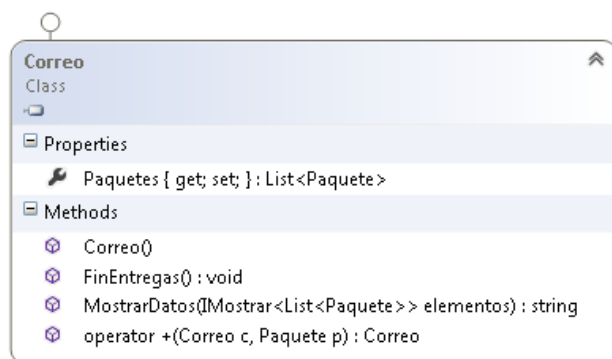
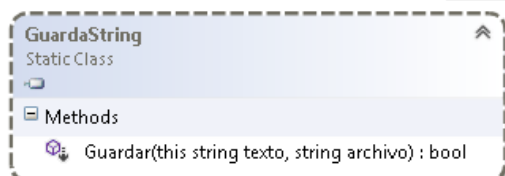
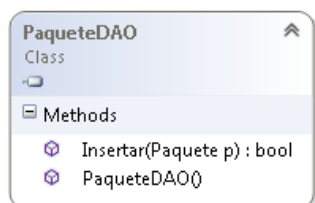
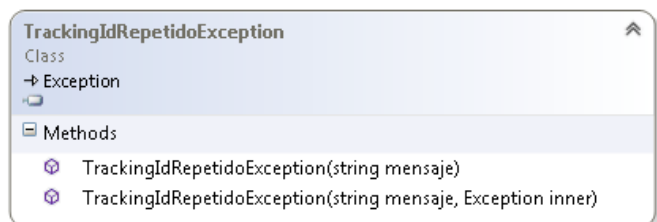
Apellido:		Fecha:	23/11/2017
Nombre:		Docente ⁽²⁾ :	H. Dillon / F. Dávila
División:	2°C	Nota ⁽²⁾ :	
Legajo:		Firma ⁽²⁾ :	
Instancia ⁽¹⁾ :	PP	RPP	SP X RSP FIN

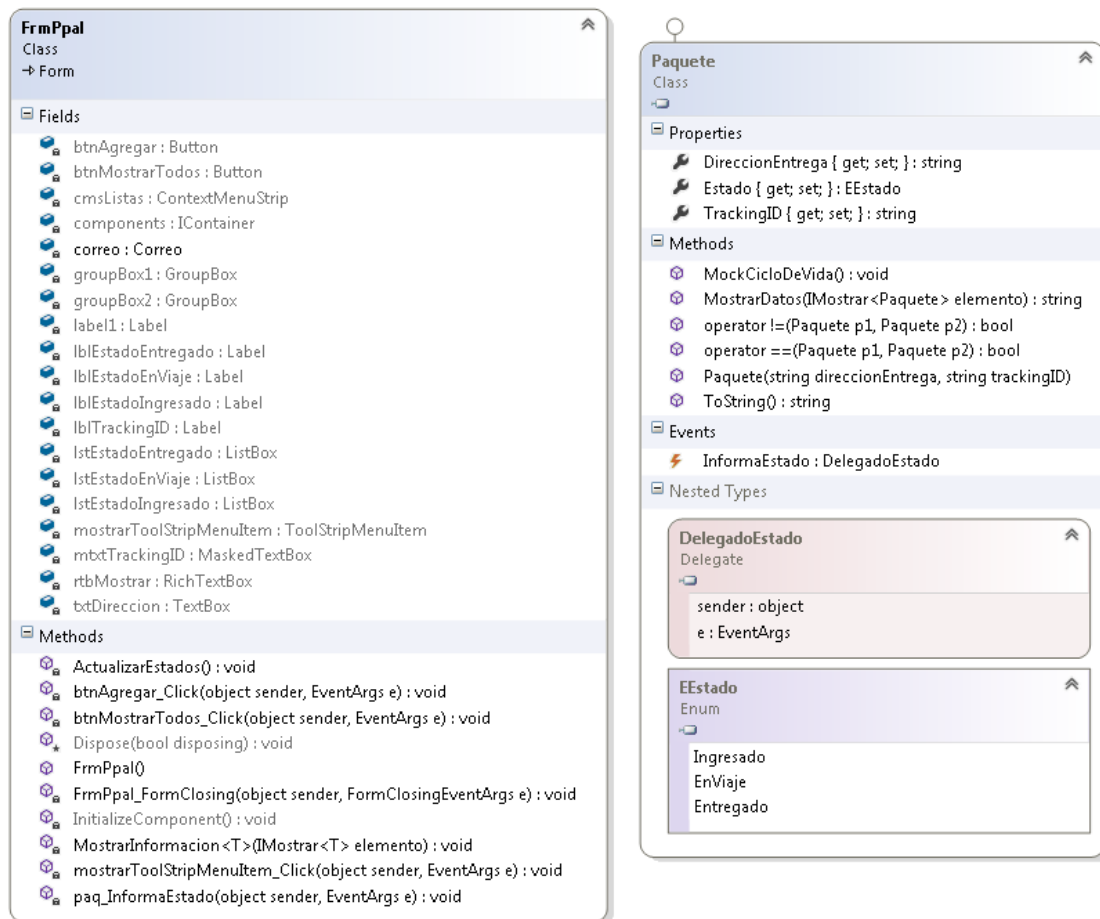
(1) Las instancias validas son: 1^{er} Parcial (PP), Recuperatorio 1^{er} Parcial (RPP), 2^{do} Parcial (SP), Recuperatorio 2^{do} Parcial (RSP), Final (FIN). Marque con una cruz.

(2) Campos a ser completados por el docente.

Pautas del examen, leer antes de iniciar el Visual Studio:

- Renombrar la carpeta pasada como maqueta: apellido.nombre.división. Por ejemplo Pérez.Juan.2E.
- Los proyectos que no sean identificables, no serán corregidos.**
- Los alumnos que no entreguen o su parcial no sea identificable serán desaprobados.**
- Sólo se corregirá lo que el alumno entregue de la siguiente forma:
 - Al finalizar, colocar la carpeta de la Solución completa en un archivo ZIP y dejar este último en el Escritorio de la máquina. Luego presionar el botón  de la barra superior, cargar un mensaje y presionar Aceptar. La barra superior deberá cambiar de color.
- En todos los casos que sea posible, reutilizar código.
- Los proyectos deberán **compilar**, sin errores.
- El proyecto de Biblioteca de Clases es un esquema, controlar y modificar lo necesario según lo pedido.
- El proyecto de Test **no debe ser modificado**, salvo en algún método específico destinado para este uso.

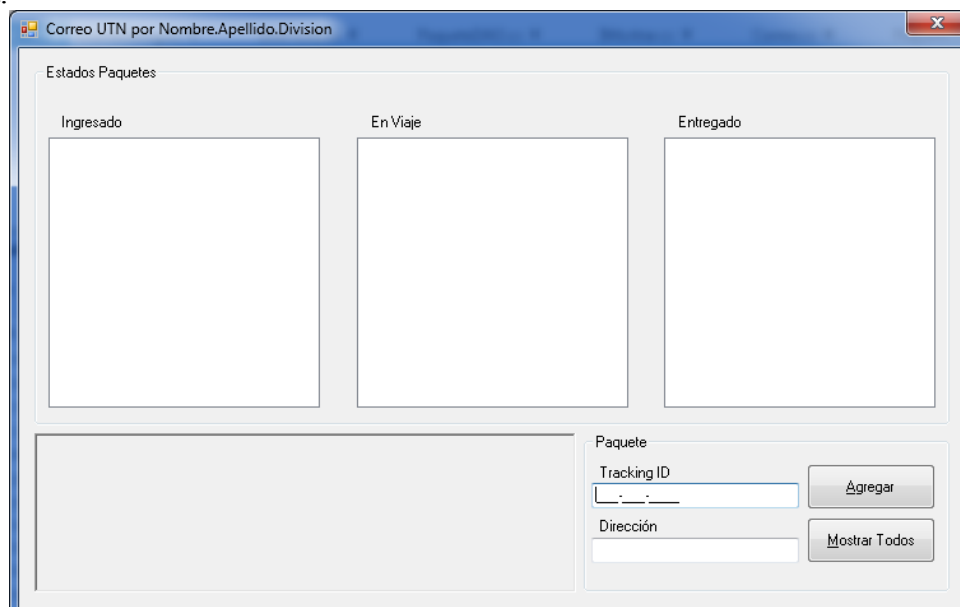




Descripción

Crearemos una aplicación para un sistema de correo.

- En la parte superior veremos los paquetes Ingresados y como cambiar su estado a En Viaje y luego Finalizados. Al alcanzar ese último estado, guardaremos la información del paquete en una base de datos provista para tal fin.
- Al seleccionar un elemento de la lista de "Entregado" y hacer click con el botón derecho del mouse, veremos un menú Mostrar. Al hacer click en este, se deberá mostrar la información del paquete en el cuadro de texto situado en la parte inferior izquierda.
- En la parte inferior derecha ingresaremos paquetes al sistema al cargar los datos y hacer click en el botón Agregar.
- Al hacer click en el botón Mostrar Todos, se mostrará la información en el cuadro de texto sito en la parte inferior izquierda y se guardará esa información en un archivo de texto en el escritorio de la máquina.



- Al proyecto provisto, agregar las referencias para que todo funcione.
- Colocar las clases pedidas en la Biblioteca de Clases ya creada.
- Dentro de las clases provistas, agregar código sólo dentro de los #region Alumno.
- Crear una base de datos con el nombre **EXACTO** "correo-sp-2017" y ejecutar el script provisto para crear una tabla dentro.

```
USE [correo-sp-2017]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[Paquetes](
    [id] [int] IDENTITY(1,1) NOT NULL,
    [direccionEntrega] [varchar](500) NOT NULL,
    [trackingID] [varchar](50) NOT NULL,
    [alumno] [varchar](50) NOT NULL,
    CONSTRAINT [PK_Paquetes] PRIMARY KEY CLUSTERED
(
    [id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_PADDING OFF
GO
```

GuardaString

1. Crear un método de extensión para la clase String
2. Este guardará en un archivo de texto en el escritorio de la máquina.
3. Recibirá como parámetro el nombre del archivo.
4. Si el archivo existe, agregará información en él.

PaqueteDAO

Clase **estática** que se encargará de guardar los datos de un paquete en la base de datos provista.

- A. De surgir cualquier error con la carga de datos, se deberá lanzar una excepción y controlarla en el método que la haya llamado, sin realizar ninguna acción con esta.
- B. El campo alumno de la base de datos deberá contener el nombre del alumno que está realizando el examen.

Paquete

1. Implementar la interfaz IMostrar, siendo su tipo genérico Paquete.
2. MostrarDatos utilizará string.Format con el siguiente formato "{0} para {1}", p.trackingID, p.direccionEntrega.
3. MockCicloDeVida hará que el paquete cambie de estado de la siguiente forma:
 - a. Colocar una demora de 10 segundos.
 - b. Pasar al siguiente estado.
 - c. Informar el estado a través de InformarEstado. EventArgs no tendrá ningún dato extra.
 - d. Repetir las acciones desde el punto A hasta que el estado sea Entregado.
 - e. Finalmente guardar los datos del paquete en la base de datos.

Correo

1. Implementará la interfaz IMostrar<List<Paquete>>.
2. En el operador +:
 - a. Controlar si el paquete ya está en la lista. En el caso de que esté, se lanzará la excepción TrackingIdRepetidoException.
 - b. Agregar el paquete a la lista de paquetes.

- c. Crear un hilo para el método MockCicloDeVida del paquete, y agregar dicho hilo a mockPaquetes.
 - d. Ejecutar el hilo.
3. MostrarDatos utilizará string.Format con el siguiente formato "{0} para {1} ({2})", p.TrackingID, p.DireccionEntrega, p.Estado.ToString() para retornar los datos de todos los paquetes de su lista.
4. FinEntregas cerrará todos los hilos activos.

Formulario

1. El evento click del botón btnAgregar realizará las siguientes acciones en el siguiente orden:
 - a. Creará un nuevo paquete
 - b. Asociará el evento InformaEstado con el método paq_InformaEstado.
 - c. Agregará el paquete al correo, controlando al menos la excepción TrackingIdRepetidoException.
 - d. Llamará al método ActualizarEstados.
2. El método MostrarInformacion:
 - a. Mostrará la información del elemento en RichTextBox rtbMostrar
 - b. Luego, utilizará el método de extensión para guardar el texto en this.rtbMostrar.Text

Test Unitarios

1. Realizar test que verifique que la lista de Paquetes del Correo esté instanciada.
2. Realizar test que verifique que no se puedan cargar dos Paquetes con el mismo Tracking ID.