





Descripción

Crearemos una aplicación para un sistema de correo, simulando los pasajes desde su ingreso hasta su entrega:

- A. 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.
 - B. 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.
 - C. En la parte inferior derecha ingresaremos paquetes al sistema al cargar los datos y hacer click en el botón Agregar.
 - D. 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.
- Crear una base de datos con el nombre **EXACTO** "correo-sp-2017" y ejecutar el script provisto a continuación 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
```

- Crear un proyecto llamado MainCorreo sólo para la vista (formulario), y otro Entidades del tipo Biblioteca de Clases.

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 generada anteriormente:

- A. De surgir cualquier error con la carga de datos, se deberá lanzar una excepción tantas veces como sea necesario hasta llegar a la vista (formulario). A través de un MessageBox informar lo ocurrido al usuario de forma clara. De ser necesario, utilizar un evento para este fin.
- B. El campo alumno de la base de datos deberá contener el nombre del alumno que está realizando el TP.

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 para compilar la información del paquete.
3. La sobrecarga del método ToString retornará la información del paquete.
4. Dos paquetes serán iguales siempre y cuando su Tracking ID sea el mismo.
5. MockCicloDeVida hará que el paquete cambie de estado de la siguiente forma:
 - a. Colocar una demora de 4 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. De no estar repetido, 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 Título del formulario deberá contener los datos del alumno, reemplazando donde dice Nombre.Apellido.Division en la imagen provista.
2. Se deberá respetar el diseño visto en la imagen, siendo los elementos a utilizar: GroupBox, ListBox, Label, RichTextBox, MaskedTextBox, TextBox, ContextMenuStrip (para el ListBox lstEstadoEntregado) y Button.
3. El evento click del botón btnAgregar realizará las siguientes acciones en el siguiente orden:
 - a. Creará un nuevo paquete y asociará al evento InformaEstado el método paq_InformaEstado.
 - b. Agregará el paquete al correo, controlando las excepciones que puedan derivar de dicha acción.
 - c. Llamará al método ActualizarEstados.
4. Al cerrarse el formulario, se deberá llamar al método FinEntregas a fin de cerrar todos los hilos abiertos.
5. paq_InformaEstado llamará al método ActualizarEstados en el ELSE del siguiente código


```
if (this.InvokeRequired)
{
    Paquete.DelegadoEstado d = new Paquete.DelegadoEstado(paq_InformaEstado);
```

```
        this.Invoke( d, new object[] {sender, e} );  
    }  
    else { // Llamar al método }
```

6. El botón btnMostrarTodos contendrá sólo la siguiente línea de código:
`this.MostrarInformacion<List<Paquete>>((IMostrar<List<Paquete>>)correo);`
7. El menú mostrarToolStripMenuItem contendrá sólo la siguiente línea de código:
`this.MostrarInformacion<Paquete>((IMostrar<Paquete>)lstEstadoEntregado.SelectedItem);`
8. El método MostrarInformacion<T> evaluará que el atributo elemento no sea nulo y:
 - a. Mostrará los datos de elemento en el rtbMostrar.
 - b. Utilizará el método de extensión para guardar los datos en un archivo llamado salida.txt.
9. El método ActualizarEstados limpiará los 3 ListBox y luego recorrerá la lista de paquetes agregando cada uno de ellos en el listado que corresponda.

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.