Controles de usuario

Apunte Nro 1

PROGRAMACIÓN II

Profesor: DARIO CARDACCI

Controles de usuario

A medida que diseña y modifica la interfaz de usuario de las aplicaciones de Windows Forms, deberá agregar, alinear y colocar los controles de usuario. Los controles son objetos contenidos dentro de objetos de formularios. Cada tipo de control tiene su propio conjunto de propiedades, métodos y eventos que lo hacen adecuado para un fin determinado. Puede manipular los controles del diseñador y escribir código para agregar controles de forma dinámica en el entorno de tiempo de ejecución.

En particular nos concentraremos en los siguientes controles aunque existen muchos más que lo invitamos a que los investigue cuando agote los que están a continuación:

Textbox. Los cuadros de texto se utilizan para obtener la entrada del usuario o para mostrar texto. El control se utiliza para texto editable, aunque también se puede convertir en de solo lectura. Los cuadros de texto pueden mostrar varias líneas, ajustar el texto al tamaño del control y agregar formato básico. El TextBox control permite un único formato para el texto que se muestra o se escribe en el control.
https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/textbox-control-windows-forms

Label. Se usan para mostrar texto o imágenes que el usuario no puede editar. Se usan para identificar objetos en un formulario, para proporcionar una descripción de lo que hará un determinado control si se hace clic en él, por ejemplo, o para mostrar información en respuesta a un evento o proceso en tiempo de ejecución de la aplicación. El control de Label no puede recibir el foco, también se puede usar para crear teclas de acceso para otros controles.
https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/label-control-windows-forms

Button. Se utiliza para colocar un botón que al oprimirlo produce que se ejecute una acción.
https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.button?view=netcore-3.1

Checkbox.	**Clase 2**
Radiobutton.	**Clase 2**
GrupBox.	**Clase 2**
Listbox.	**Clase 2**
Combobox.	**Clase 2**
Monthcalendar.	**Clase 2**
ProgressBar.	**Clase 2**
DataGridView.	**Clase 2**
ColorDialog.	**Clase 2**
FontDialog.	**Clase 3**
Linklabel.	**Clase 3**
Checkedlistbox.	**Clase 3**

DateTimePicker. Clase 3
ListView. Clase 3
NotifyIcon. Clase 3
NumericUpDown. Clase 3
PictureBox. Clase 3
RichtextBox. Clase 3
ToolTip. Clase 4
TreeView. Clase 4
WebBrowser. Clase 4
MenuStrip. Clase 4
FolderBrowseDialog. Clase 4
OpenFileDialog. Clase 4
SaveFileDialog. Clase 4
PrintDialog. Clase 4

Checkbox:

Esta clase nos permite crear un checkbox dentro de un formulario windows. La podemos encontrar dentro del C# Designer, o crearlo mediante:

```
public void InstantiateMyCheckBox()
 {
   // Create and initialize a CheckBox.
   CheckBox checkBox1 = new CheckBox();

   // Make the check box control appear as a toggle button.
   checkBox1.Appearance = Appearance.Button;

   // Turn off the update of the display on the click of the control.
   checkBox1.AutoCheck = false;

   // Add the check box control to the form.
   Controls.Add(checkBox1);
 }
```

Dentro de los eventos interesante que tiene esta clase son:

- CheckedChanged: Se dispara cuando cambio el estado del checkbox
- DoubleClick: Ocurre al hacer dos clicks en el checkbox

Radiobutton

Genera un radiobutton(o varios) donde solo se puede seleccionar una opción. Se puede crear a través del C# Designer, o crearlo mediante:

```
private GroupBox groupBox1;
private RadioButton radioButton2;
private RadioButton radioButton1;

public void InitializeRadioButtons()
{
```

```
    this.groupBox1 = new System.Windows.Forms.GroupBox();
    this.radioButton2 = new System.Windows.Forms.RadioButton();
    this.radioButton1 = new System.Windows.Forms.RadioButton();

    this.groupBox1.Controls.Add(this.radioButton2);
    this.groupBox1.Controls.Add(this.radioButton1);
}
```

Nótese que creamos un objeto GroupBox ya que este se va a encargar de contener los objetos Radiobutton(ya que tenemos distintas opciones y están conectadas)

Groupbox

Genera un groupbox que nos sirve como marco(o frame) donde contener otros tipos de controles(por ejemplo radiobuttons). Se puede crear a traves del C# Designer, o crearlo mediante:

```
private void InitializeMyGroupBox()
{
    // Create and initialize a GroupBox and a Button control.
    GroupBox groupBox1 = new GroupBox();
}
```

Para sumar un objeto dentro del frame tenemos que usar el método Add

```
    // Add the Button to the GroupBox.
    groupBox1.Controls.Add(FooMyObject);
```

Listbox

Nos permite generar un listado de objetos que el usuario va a poder clickear. Lo podemos crear mediante:

```
// Create an instance of the ListBox.
   ListBox listBox1 = new ListBox();
```

Luego para insertar items dentro de la colección podemos usar el método Items.Add. Nótese que primero llamamos al método BeginUpdate() y luego EndUpdate() para inicializar/finalizar la transacción:

```
   // Shutdown the painting of the ListBox as items are added.
   listBox1.BeginUpdate();
   // Loop through and add 50 items to the ListBox.
   for (int x = 1; x <= 50; x++)
   {
      listBox1.Items.Add("Item " + x.ToString());
   }
   // Allow the ListBox to repaint and display the new items.
   listBox1.EndUpdate();
```

Podemos insertar ítems de forma directa(sin Begin/End) pero el objeto va a estar recargando el display de ítems todo el tiempo. Si usamos el método anterior se van a mostrar los ítems una vez finalizada la escritura.

Combobox

El combobox nos permite mezclar un ListBox y un TextBox. Esta clase genera algo muy similar a un ListBox pero con un estilo *DropDown.* Podemos instanciar uno usando:

```
private System.Windows.Forms.ComboBox comboBox1;
```

Dentro de esta clase podemos usar la misma lógica para agregar ítems mencionada en ListBox haciendo uso de Items.add y Begin/End update.

Contamos con el método FindIndex("FooMyText") que nos va a permitir encontrar el indice de un ítem dentro de una lista mediante un texto. Luego seleccionamos ese ítem mediante la propiedad SelectedIndex.

```
int index = comboBox1.FindString(textBox2.Text);
comboBox1.SelectedIndex = index;
```

Notese que el método FindIndex no es case-sensitive por lo tanto no tendrá en cuenta las mayúsculas/minúsculas a la hora de buscar un ítem

MonthCalendar

Esta clase nos permite crear un calendario.

```
// Create the calendar.
this.monthCalendar1 = new System.Windows.Forms.MonthCalendar();
```

A través de distintas propiedades podemos setear la fecha máxima, fecha minima, formatos, etc. Ejemplo:

```
// Set week to begin on Monday.
this.monthCalendar1.FirstDayOfWeek = System.Windows.Forms.Day.Monday;

// Set the maximum visible date on the calendar to 12/31/2010.
this.monthCalendar1.MaxDate = new System.DateTime(2010, 12, 31, 0, 0, 0, 0);

// Set the minimum visible date on calendar to 12/31/2010.
this.monthCalendar1.MinDate = new System.DateTime(1999, 1, 1, 0, 0, 0, 0);
```

Un evento util es el DateSelected que se dispara cuando termino de seleccionar la fecha final(luego de seleccionar la fecha inicial) para acceder a la fecha seleccionada por el usuario.

Para manipular las fechas seleccionadas podemos apoyarnos en esta función, donde la fecha viene como argumento del evento("e")

```
private void monthCalendar1_DateSelected(object sender,
System.Windows.Forms.DateRangeEventArgs e)
   {
```

```
        // Show the start and end dates in the text box.
        this.textBox1.Text = "Date Selected: Start = " +
            e.Start.ToShortDateString() + " : End = " + e.End.ToShortDateString();
    }
```

Progress Bar

Nos permite generar barras de carga. Estas barras pueden ser de dos formatos:

- Bloques segmentados
- Barra continua que se va llenando
- Bloque que se desplaza por el espacio de barra

El mínimo y el maximo de la barra representan un valor numero(asignado por nosotros) que va a ir aumentando en pasos definidos por nosotros también. Este funcionamiento permite que ajustemos la escala a nuestro gusto(tanto para escalas lineales o no lineales).

Las propiedades mencionadas son:

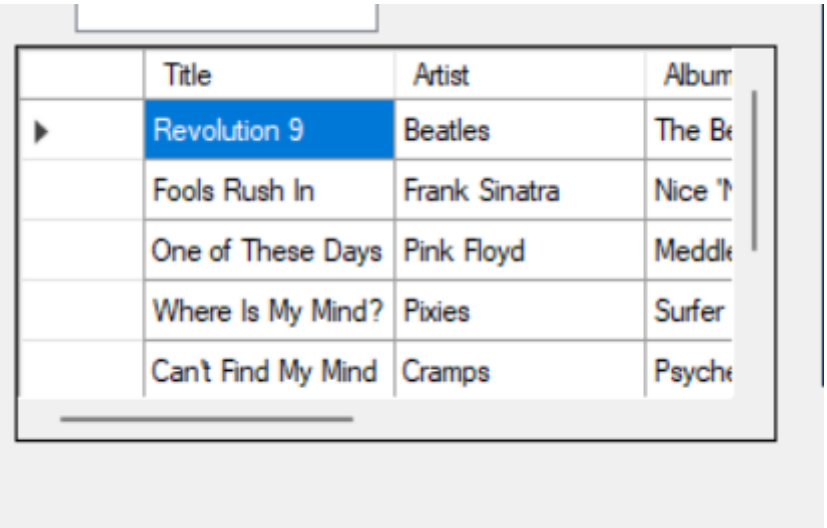
```
// Display the ProgressBar control.
    pBar1.Visible = true;
    // Set Minimum to 1 to represent the first file being copied.
    pBar1.Minimum = 1;
    // Set Maximum to the total number of files to copy.
    pBar1.Maximum = filenames.Length;
    // Set the initial value of the ProgressBar.
    pBar1.Value = 1;
    // Set the Step property to a value of 1 to represent each file being copied.
    pBar1.Step = 1;
```

El codigo de los controles se encuentra en la solucion Clase 3 y Clase 4

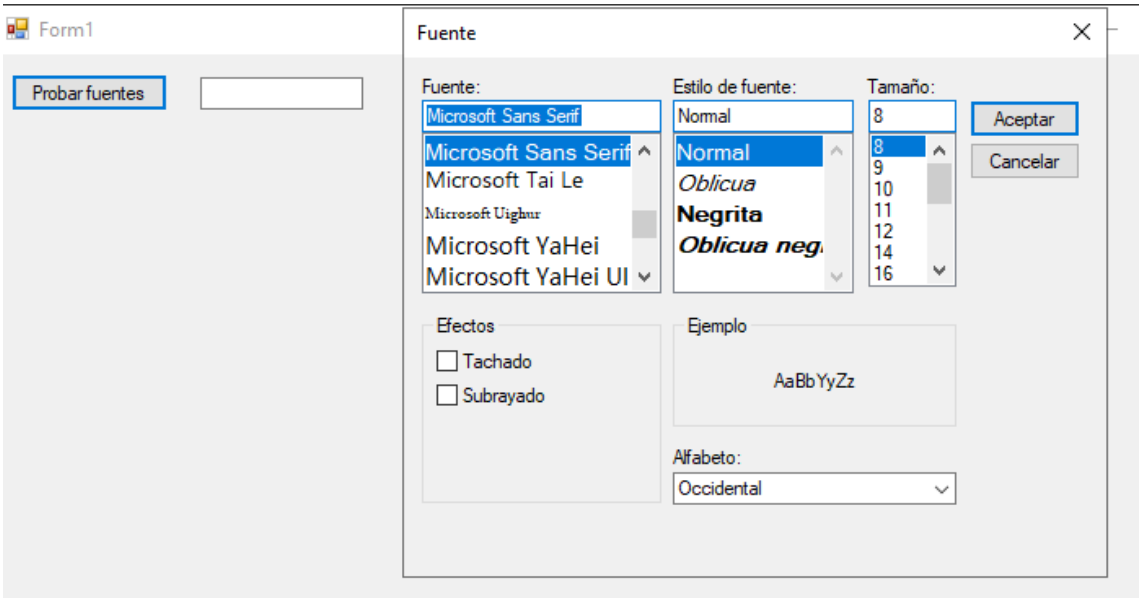
DataGridView

—----------

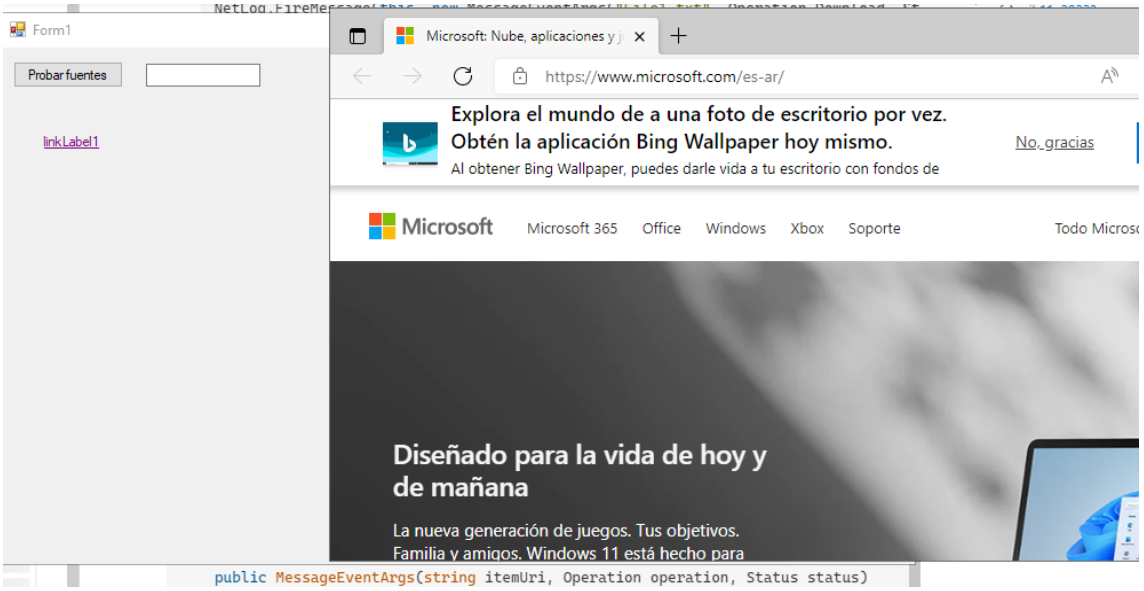
Podemos genera un contenedor para mostrar data de forma mas comoda


Title
Artist
Album
Revolution 9
Beatles
The Be
Fools Rush In
Frank Sinatra
Nice 'N
One of These Days
Pink Floyd
Meddle
Where Is My Mind?
Pixies
Surfer
Can't Find My Mind
Cramps
Psyche

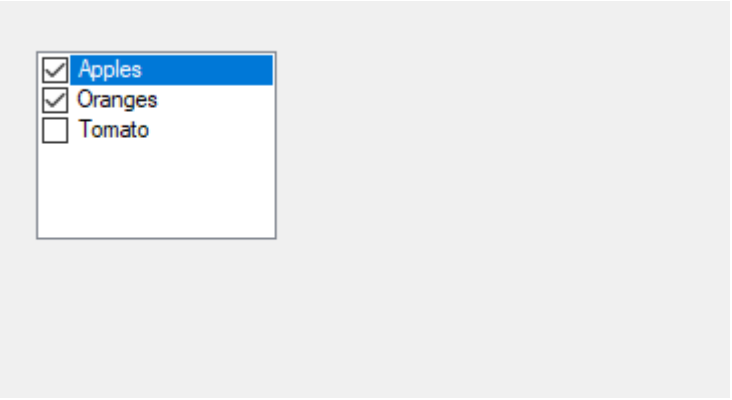
FONT Dialog


Form1
Probar fuentes
Fuente
Fuente:
Microsoft Sans Serif
Microsoft Sans Serif
Microsoft Tai Le
Microsoft Uighur
Microsoft YaHei
Microsoft YaHei UI
Estilo de fuente:
Normal
Normal
Oblicua
Negrita
Oblicua neg.
Tamaño:
8
8
9
10
11
12
14
16
Aceptar
Cancelar
Efectos
Tachado
Subrayado
Ejemplo
AaBbYyZz
Alfabeto:
Occidental

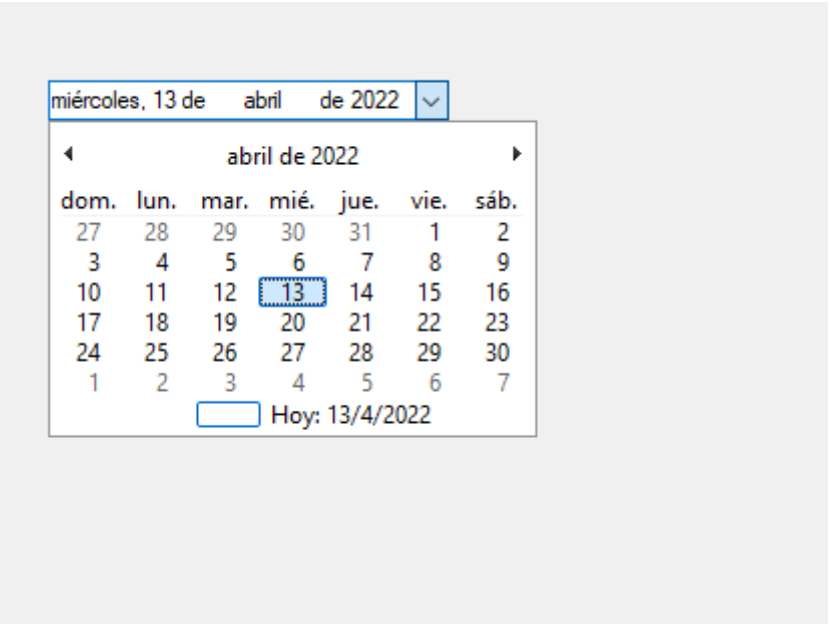
LinkLabel



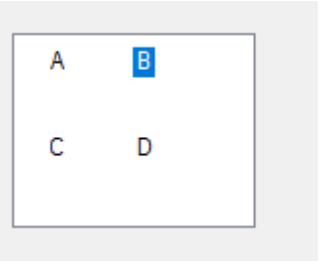
CheckedListBox



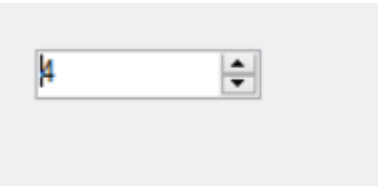
DatetimePicker



ListView

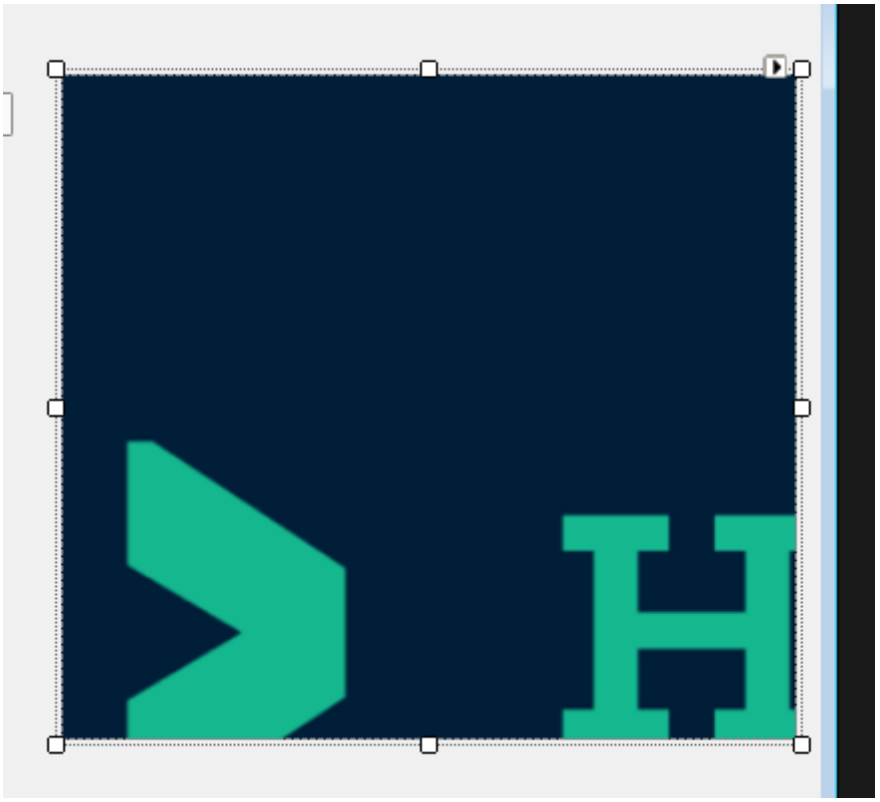


NumericUpDown:



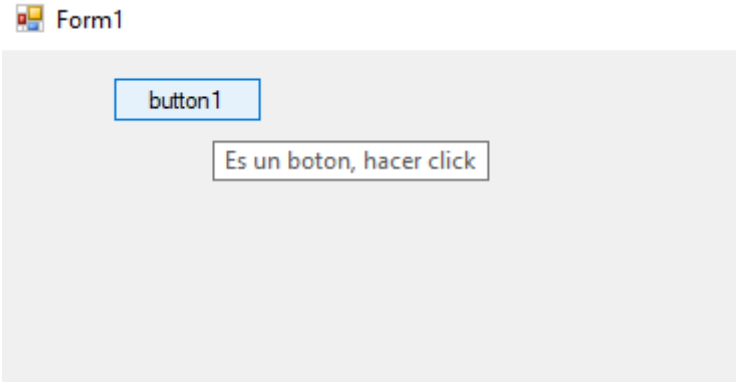
PictureBox

PIERRI, Matias Gabriel ININF2

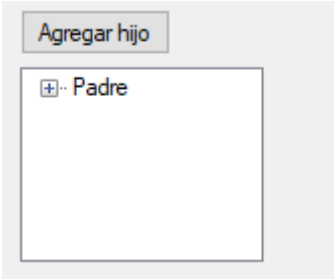


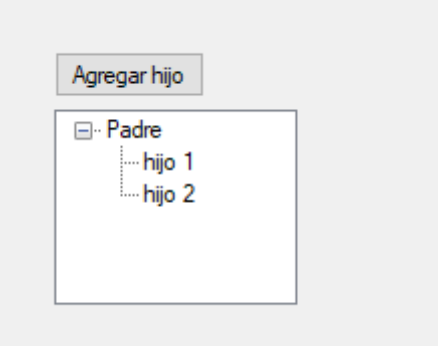
—----------

ToolTip



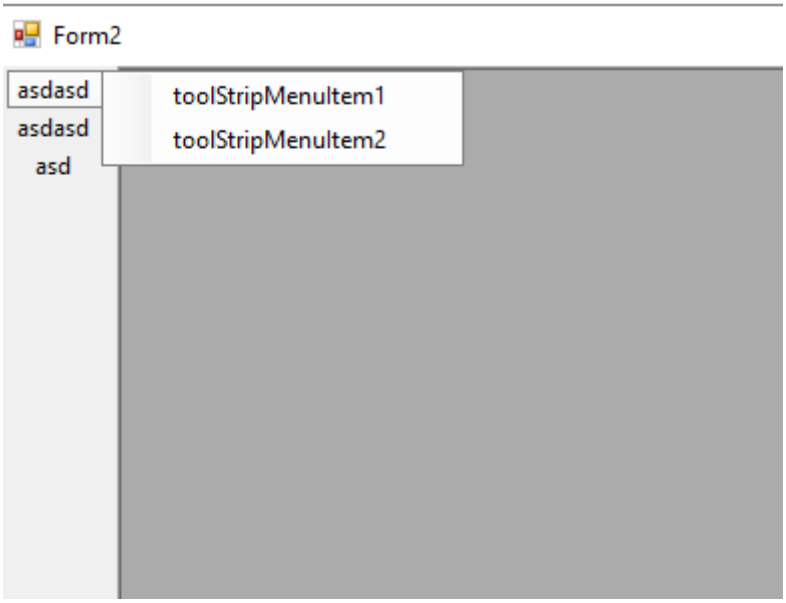
TreeView



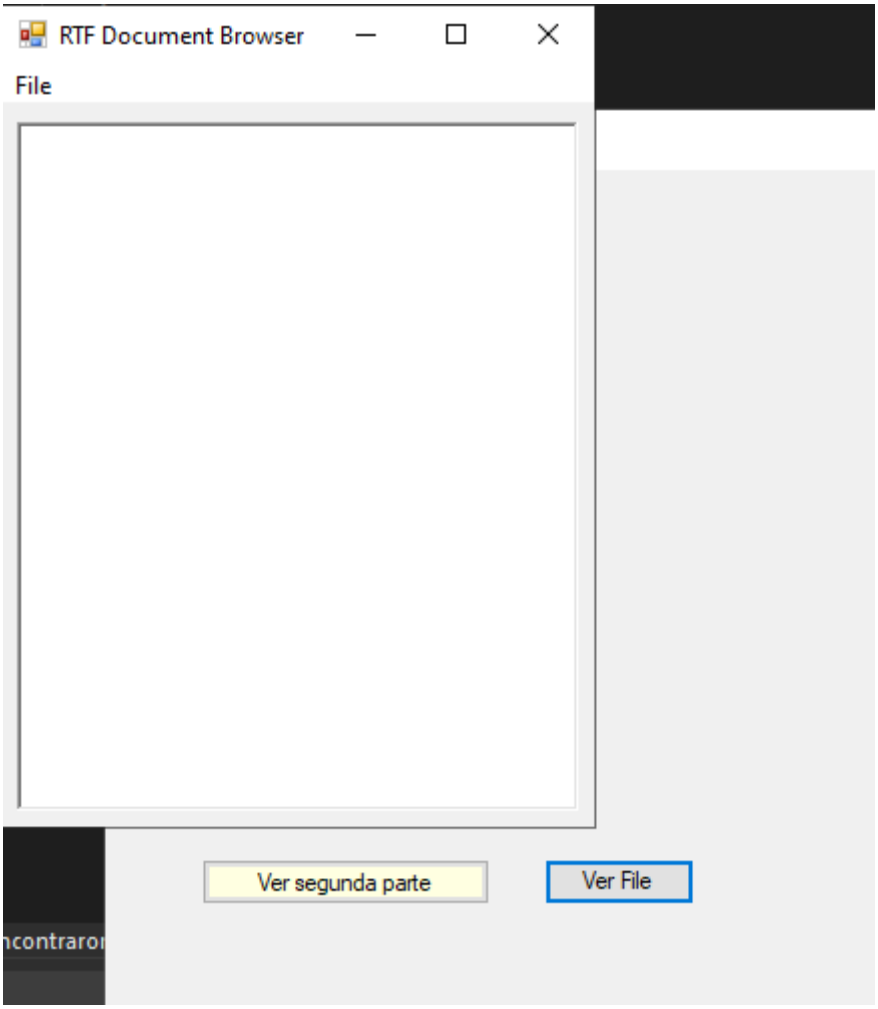


WebBrowser

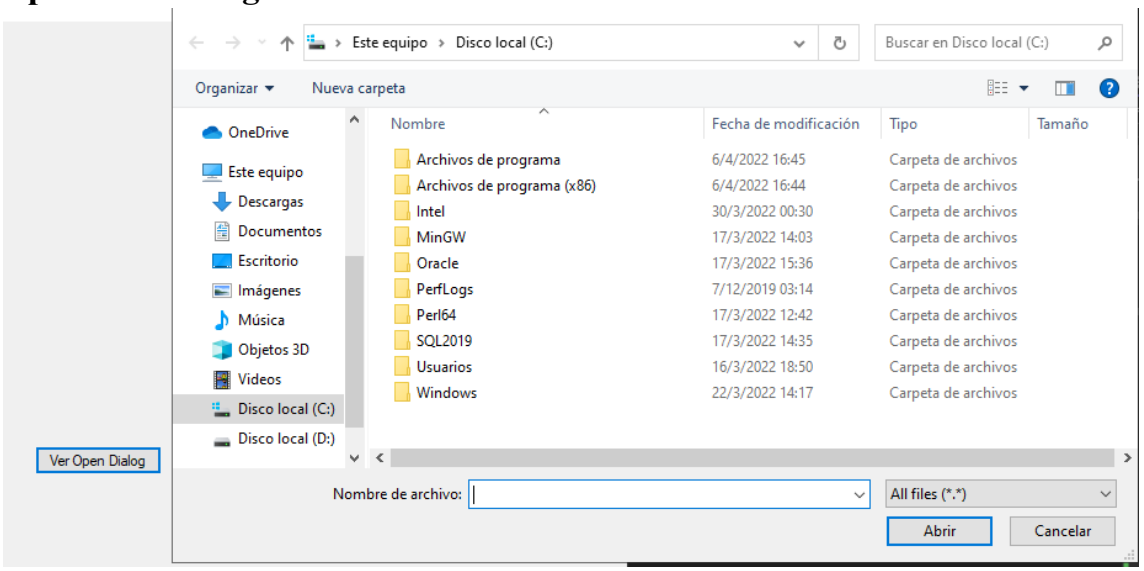
MenuStrip

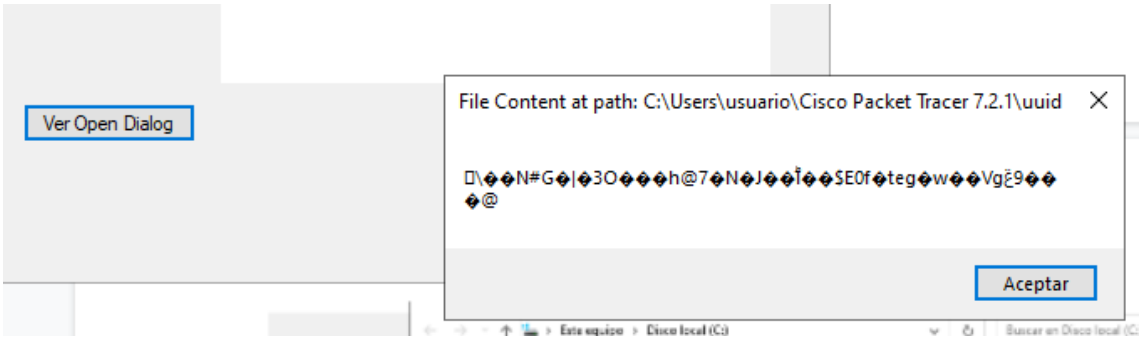


FolderBrowseDialog

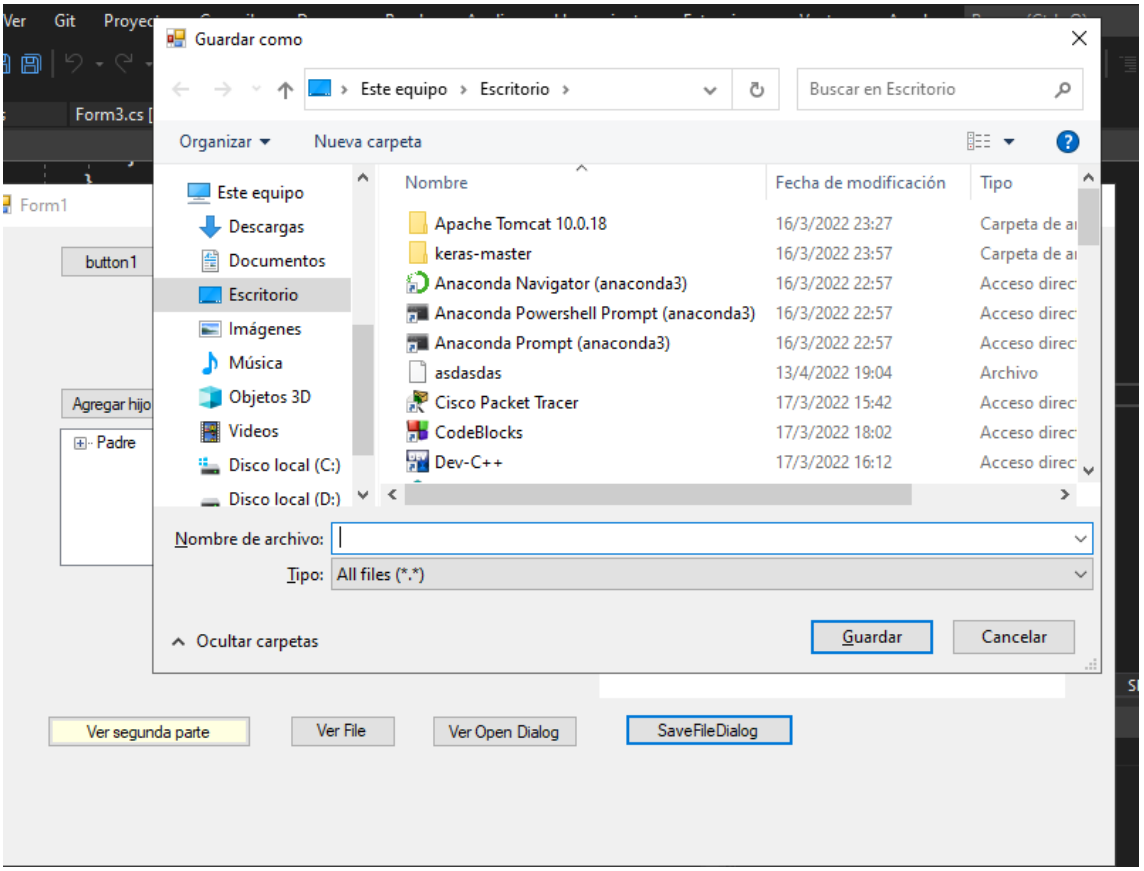


Open File Dialog

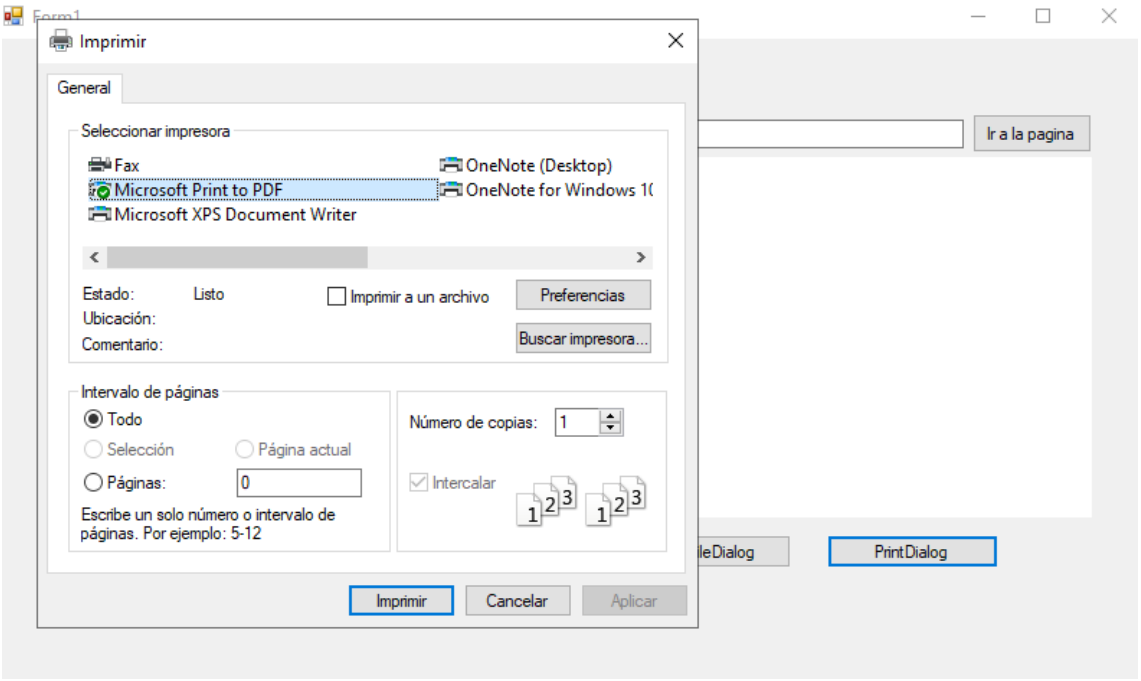




Save Dialog



Print Dialog



Tipos de Datos y Variables

https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/types-and-variables

Arrays

https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/arrays/

Estructuras de decisión

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/if-else

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/switch

Estructuras de repetición

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/do

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/for

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/foreach-in

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/while

Instrucciones de salto

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/break

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/continue

https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/return