

Controles III

FontDialog
ColorDialog
OpenFileDialog
SaveFileDialog*
Personalizadas

Cajas de Diálogo Estándar.

Introducción

- La biblioteca .NET incluye una serie de clases, derivadas todas ellas de **CommonDialog** y definidas en el espacio de nombres **System.Windows.Forms**, que permiten visualizar las cajas de diálogo más comúnmente empleadas en el diseño de aplicaciones, tales como: Abrir o Guardar y la cja de diálogo Color o fuente... Así como otras definidas por el usuario.
- Pasemos a verlas...

FontDialog

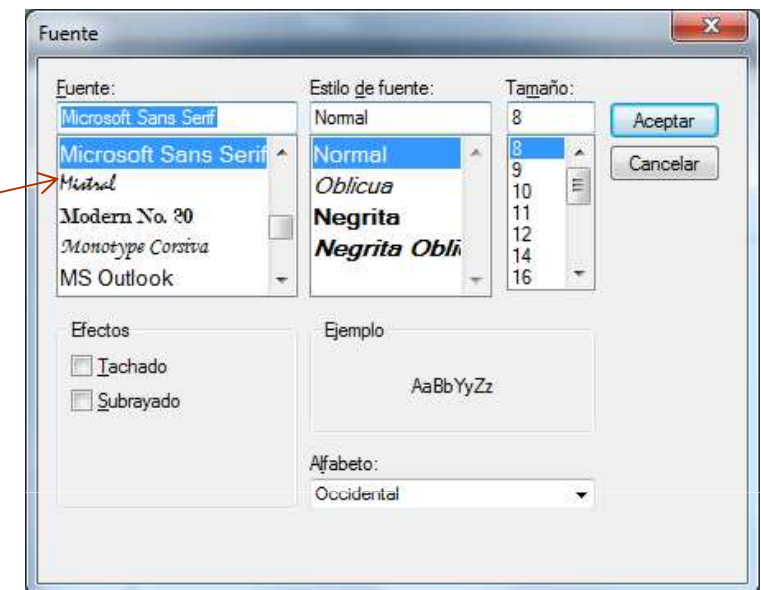
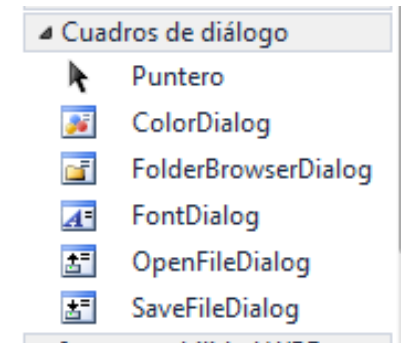
Vamos a cambiar la fuente a un determinado elemento

En una aplicación windows añadimos un botón, un textbox y fontdialog, en el clic del botón escribimos el siguiente código:

```
private void button1_Click(object sender, EventArgs e)
{
    textBox1.Text = "Cambiar fuente";

    fontDialog1.ShowDialog();

    textBox1.Font = fontDialog1.Font;
    textBox1.ForeColor = fontDialog1.Color;
}
```



Hay una serie de propiedades que nos permite mostrar u ocultar elementos en el cuadro de diálogo

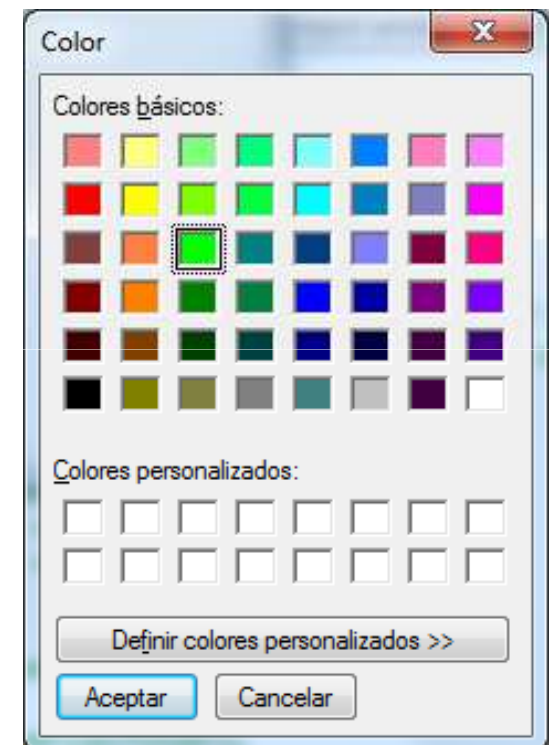
fontDialog1.ShowColor=true/false;

fontDialog1.ShowEffects=true/false;

fontDialog1.ShowHelp = true/false;

ColorDialog

- `colorDialog1.Color = textBox1.BackColor;`
- `colorDialog1.ShowDialog();`
- `textBox1.BackColor = colorDialog1.Color;`



OpenFileDialog

- Veamos algunas propiedades de este control
 - **openfiledlg1.Filter** // Definimos los filtros de archivos a permitir.
 - `openfiledlg1.Filter = "Bitmap files (*.bmp) | *.bmp | Gif files (*.gif) | *.gif | JGP files (*.jpg) | *.jpg | All (*.*) | *.* | PNG (*.patito) | *.png";`
 - `openfiledlg1.Filter = "ficheros txt (*.txt) | *.txt | Todos (*.*) | *.*";`
 - **openfiledlg1.FilterIndex = 3;** // Establece que filtro se mostrará por defecto en este caso, 3=jpg
 - **Openfiledlg1.Title** // El título de la Ventana....
 - `openfiledlg1.Title = "Ficheros 1DAW-SM";`
 - **openfiledlg1.InitialDirectory** = "c:\\\\"; // El directorio que por defecto habrá, para cada contrapleca del Path colocar \\

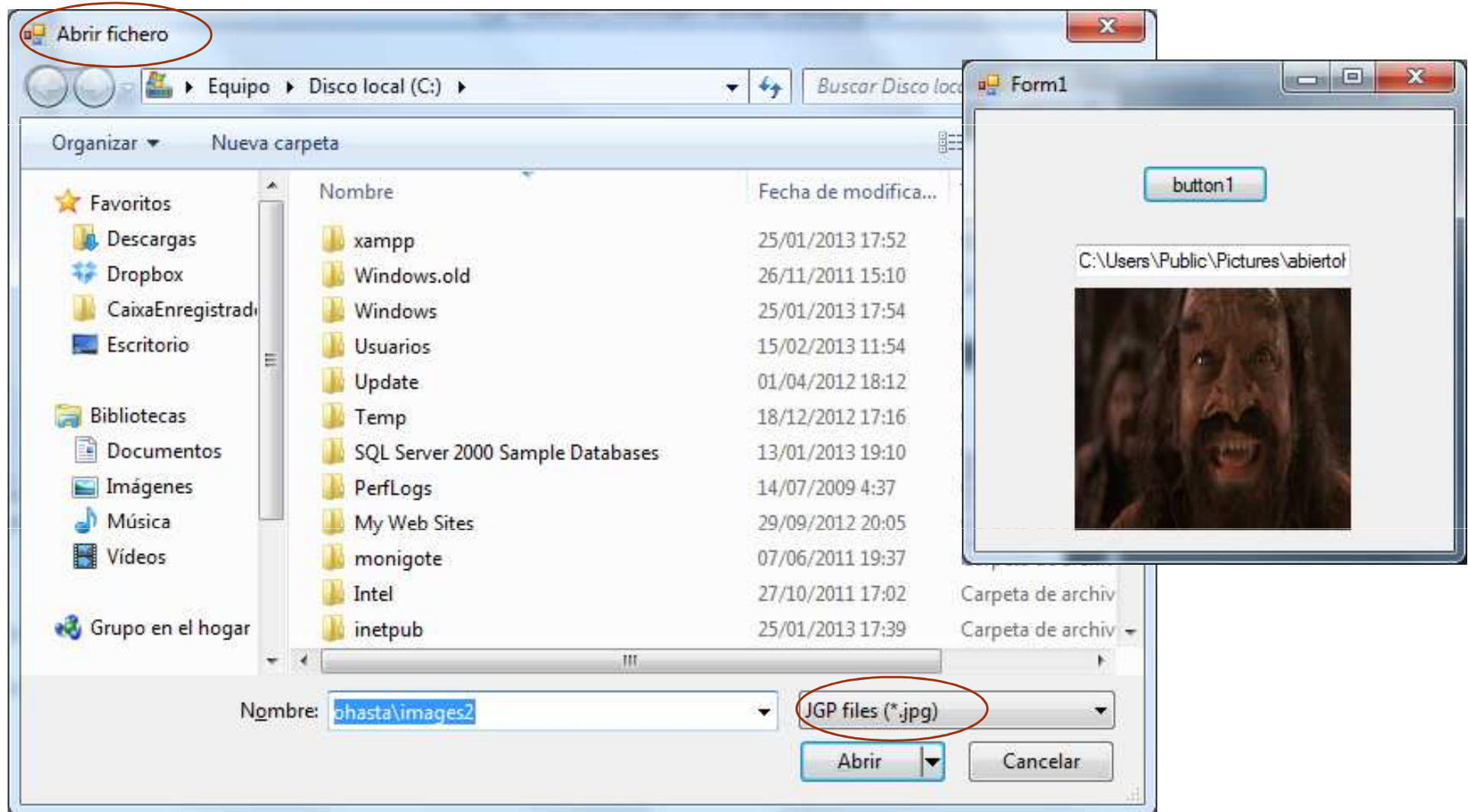
OpenFileDialog

```
//Definimos los filtros de archivos a permitir.  
  
//openFileDialog1.Filter = "ficheros txt (*.txt)|*.txt|Todos (*.*)|*.*";  
openFileDialog1.Filter = "Bitmap files (*.bmp)|*.bmp|Gif files (*.gif)|*.gif|JGP files (*.jpg)|*.jpg|All (*.*)|*.* |PNG (*.patito)|*.png ";  
  
openFileDialog1.FilterIndex = 3;  
//El titulo de la Ventana....  
openFileDialog1.Title = "Abrir fichero";  
openFileDialog1.InitialDirectory = "c:\\\\";  
if (openFileDialog1.ShowDialog() == DialogResult.OK)  
{  
    string fitxer;  
    fitxer = openFileDialog1.FileName;  
    textBox1.Text = fitxer;  
    pictureBox1.Image = Image.FromFile(fitxer);  
}
```

El valor de tipo **DialogResult** devuelto por **ShowDialog** indica si el usuario aceptó la operación (**OK**) o la canceló (**Cancel**)

Cargar una imagen desde una ruta..

OpenFileDialog: El resultado



SaveFileDialog

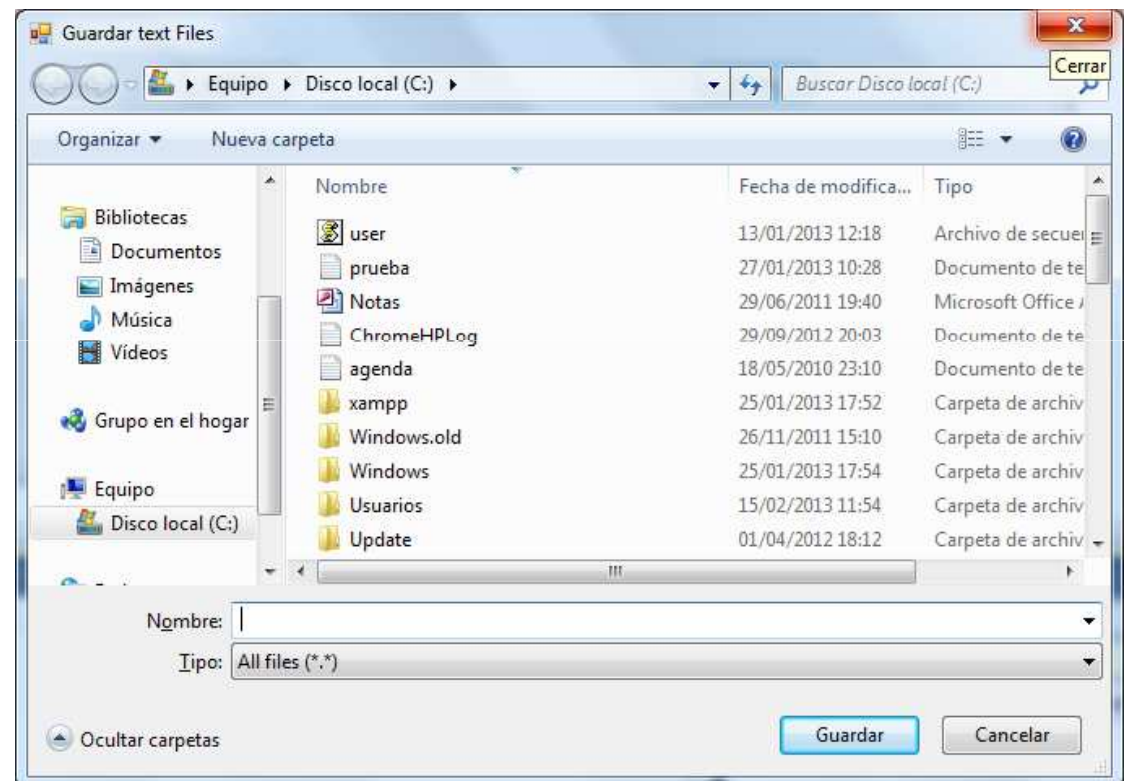
- Es idéntico al OpenFileDialog...

```
saveFileDialog1.InitialDirectory = @"C:\";  
saveFileDialog1.Title = "Guardar text Files";  
saveFileDialog1.CheckFileExists = true;  
saveFileDialog1.CheckPathExists = true;  
saveFileDialog1.DefaultExt = "txt";  
saveFileDialog1.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";  
saveFileDialog1.FilterIndex = 2;
```

```
if (saveFileDialog1.ShowDialog() == DialogResult.OK)  
{  
    textBox1.Text = saveFileDialog1.FileName;  
}
```

Un comentario con respecto al ejemplo anterior:
La @ delante nos permite utilizar la nomenclatura estándar

- ...Se verá, más cuando veamos ficheros.



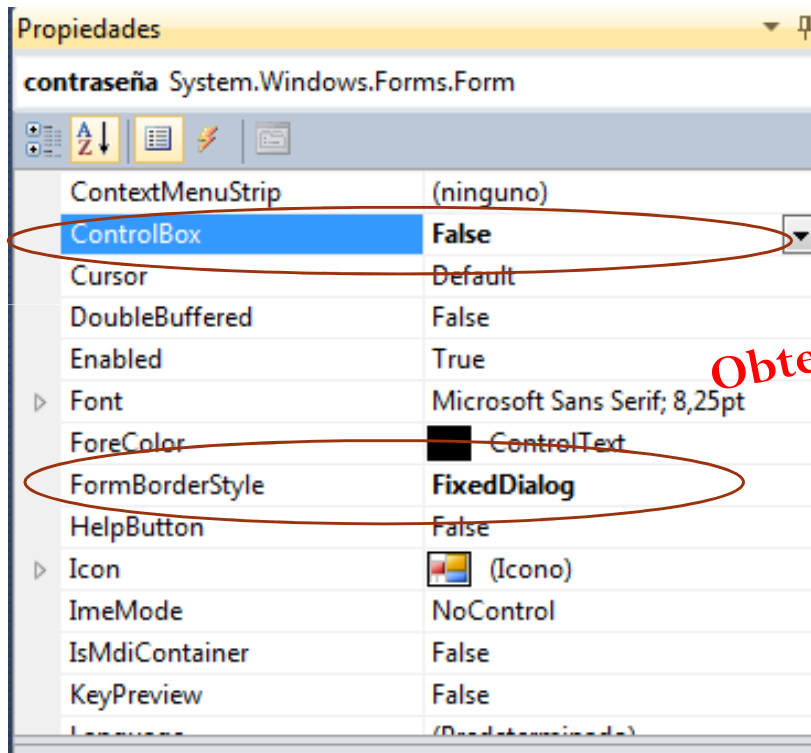
Cajas de Diálogo Personalizadas.

Introducción

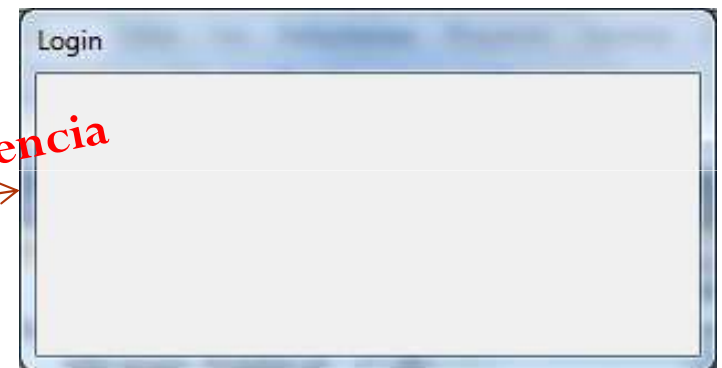
- Además de los diálogos predefinidos, podemos crear nuestras propias ventanas de *diálogo personalizadas* mediante el diseñador de formularios.
- Basta con crear un form y crear controles como **Label**, **TextBox** o **Button**, entre otros para adaptar el diálogo a nuestras necesidades

Cajas de Diálogo Personalizadas. Crear

1. Agregar un formulario al proyecto.
2. Las cajas de diálogo no suelen incluir barras de menús, de desplazamiento de ventana, botones de minimizar y maximizar, ni borde de tamaño ajustable, por lo tanto, en la ventana propiedades cambiar **FormBorderStyle** a *FixedDialog* y **ControlBox** a *false*

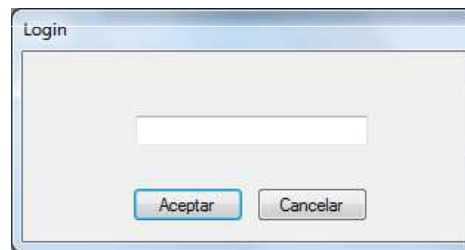


Obtenemos esta apariencia



Cajas de Diálogo Personalizadas. Crear

- Vamos a personalizarla...
- Crearemos dos botones y un TextBox...



- ...E iremos a las propiedades y le asignamos...

AutoEllipsis	False
AutoSize	None
AutoSizeMode	OK
BackColor	Cancel
BackgroundImage	Abort
BackgroundImageLayout	Retry
CausesValidation	Ignore
ContextMenuStrip	Yes
Cursor	No
DialogResult	OK
Dock	None

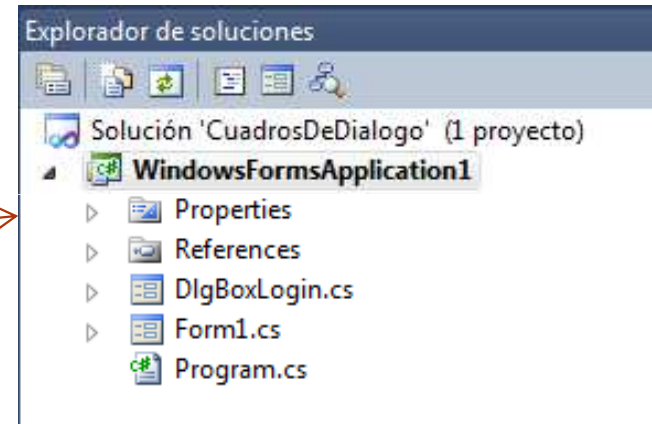
AutoEllipsis	raise
AutoSize	None
AutoSizeMode	OK
BackColor	Cancel
BackgroundImage	Abort
BackgroundImageLayout	Retry
CausesValidation	Ignore
ContextMenuStrip	Yes
Cursor	No
DialogResult	Cancel
Dock	None

- ...respectivamente.

Cajas de Diálogo Personalizadas.

Invocar

Tenemos la siguiente solución, donde vemos 2 formularios, Form1.cs y DlgBoxLogin.cs como invoco DlgBoxLogin.cs desde Form1.cs...



```
private void button3_Click(object sender, EventArgs e)
{
    DlgBoxLogin dlg = new DlgBoxLogin();
    dlg.ShowDialog();
}
```

...Pues muy sencillo, creando un objeto de su clase (**new**) DlgBoxLogin.cs y mostrarlo (**Show o ShowDialog**)

Cajas de Diálogo Personalizadas.

Recuperar datos

- Para recuperar los datos es importante saber como se cerró: valor **DialogResult** devuelto por **ShowDialog**. Por ejemplo, si el usuario clic en *Aceptar* los datos se recuperarán, pero si lo hacen en *Cancelar*, los datos escritos por el usuario se desecharán.

```
DlgBoxLogin dlg = new DlgBoxLogin();  
|  
if (dlg.ShowDialog() == DialogResult.OK)  
    MessageBox.Show(dlg.textBox1.Text);  
else  
    MessageBox.Show("Cerrar");
```

Hace referencia al
TextBox del cuadro de dialogo

...Si cuando llamamos a un elemento, textbox o cualquier otro, de la clase que invocamos y no nos aparece en la lista e problema puede ser, por que el modificador del elemento está como **private**.

Cambiarlo a **internal** o **public**

