

Guía 2 de ejercicios en C#

ING. ORLANDO GIRON / TEC. SAMUEL RODRIGUEZ

Programación 1

UNIVERSIDAD TECNOLÓGICA DE EL SALVADOR | FACULTAD DE INFORMÁTICA Y CIENCIAS
APLICADAS

CAJAS DE MENSAJES

En este ejemplo vamos a estudiar el método Show de la clase MessageBox. Dicho método es polimorfo y puede mandarse llamar de 21 diferente maneras. La que veremos aquí tiene la forma:

```
DialogResult MessageBox.Show(string text, string caption,  
MessageBoxButtons buttons, MessageBoxIcon icon);
```

La primera cadena (text), se despliega como mensaje en una caja de dialogo.

La segunda cadena (caption), se despliega como título de la caja de dialogo o ventana

buttons es un elemento de la enumeración MessageBoxButtons que puede tomar uno de seis diferentes valores y que dependiendo de ello mostrará 1, 2 o 3 botones en la caja de dialogo:

- OK
- OKCancel
- YesNo
- YesNoCancel
- RetryCancel
- AbortRetryIgnore

icon es un elemento de la enumeración MessageBoxIcon que puede tomar uno de cuatro diferentes valores y que dependiendo de ello mostrará el icono correspondiente:

- Information
- Exclamation
- Question
- Error

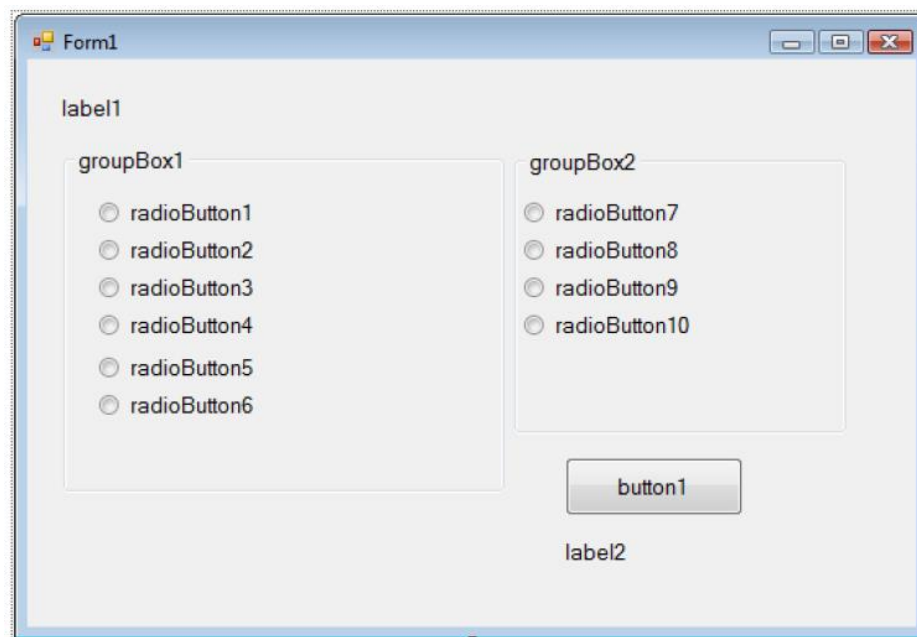
El método Show de MessageBox siempre nos regresará un resultado del tipo DialogResult y éste dependerá del botón que oprimamos al cerrar el dialogo de Mensaje. DialogResult es una enumeración que tiene los siguientes valores:

- DialogResult.OK
- DialogResult.Cancel
- DialogResult.Yes
- DialogResult.No
- DialogResult.Ignore
- DialogResult.Abort
- DialogResult.Retry

Crearemos un nuevo proyecto al que llamaremos MessageBoxes. En la forma Form1 generada, depositaremos:

- 2 Etiquetas (Label)
- 2 Cajas (GroupBox)
- 1 Botón (Button)
- 10 Botones de radio (RadioButton)

Quedará de la siguiente manera:



Cambiamos las propiedades según se muestra en la siguiente tabla, la propiedad Name de cada elemento no se cambia:

Componente	Propiedad	Valor
Form1	Text	Ejemplos MessageBox
label1	Text	Elige el MessageBox para visualizarlo
label2	Name	label2
	Text	== nada ==
button1	Text	Visualizar
groupBox1	Text	Tipo de botón
groupBox2	Text	Tipo de icono
radioButton1	Text	OK
	Checked	True
radioButton2	Text	OK Cancel
radioButton3	Text	Yes No
radioButton4	Text	Yes No Cancel
radioButton5	Text	Retry Cancel
radioButton6	Text	Abort Retry Cancel
radioButton7	Text	Information
	Checked	True
radioButton8	Text	Exclamation
radioButton9	Text	Question
radioButton10	Text	Error

Ejemplos MessageBox

Elige el MessageBox para visualizarlo

Tipo de Botón

- ☒ Ok
- ☐ Ok Cancel
- ☐ Yes No
- ☐ Yes No Cancel
- ☐ Retry Cancel
- ☐ Abort Retry Cancel

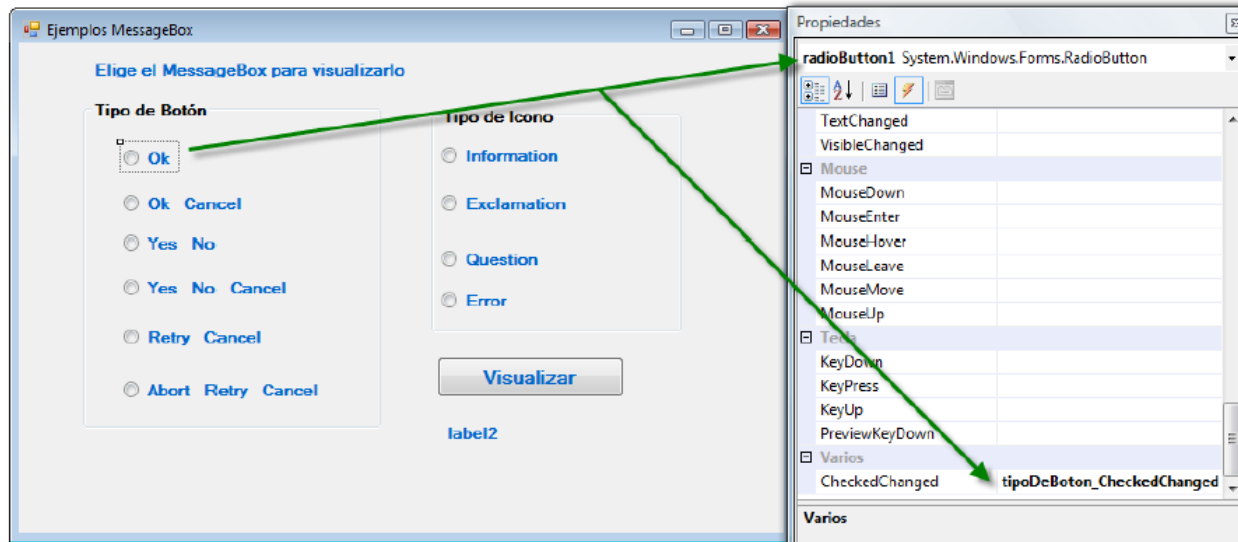
Tipo de Icono

- ☒ Information
- ☐ Exclamation
- ☐ Question
- ☐ Error

Visualizar

label2

Ahora programemos el evento para que se genere un cambio en la sección Tipo de Botón del lado izquierdo, teniendo seleccionado el radioButton1 con etiqueta OK, hagamos clic en el botón que tiene un rayo en la ventana de propiedades (para seleccionar eventos) y seleccionemos CheckedChange, a la derecha escribamos como nombre para el método que se va a ejecutar cuando este evento se lleve a cabo, tipoDeBoton_CheckedChange, hagamos lo mismo para los otros cinco radioButtons restantes seleccionando siempre el mismo método.



Lo que sigue es agregar una variable privada a la clase y lo hacemos antes del constructor:

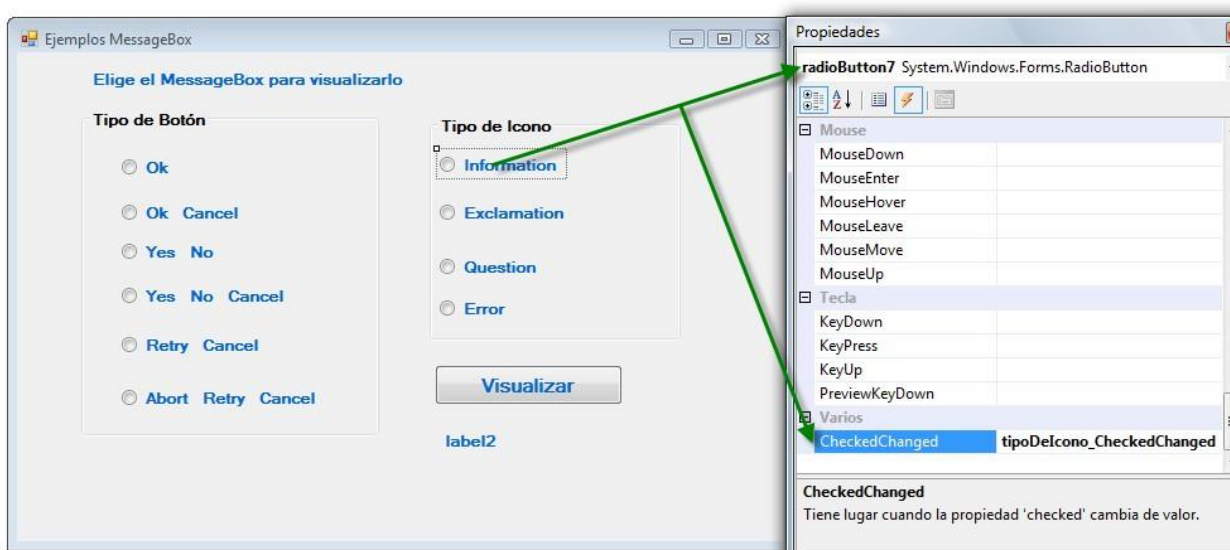
```
private MessageBoxButtons tipoDeBoton = MessageBoxButtons.OK;
```

Luego insertemos el siguiente código para el método tipoDeBoton_CheckedChanged

```
private void tipoDeBoton_CheckedChanged(object sender, EventArgs e)
{
    if (sender == radioButton1)
        tipoDeBoton = MessageBoxButtons.OK;
    else if (sender == radioButton2)
        tipoDeBoton = MessageBoxButtons.OKCancel;
    else if (sender == radioButton3)
        tipoDeBoton = MessageBoxButtons.YesNo;
    else if (sender == radioButton4)
        tipoDeBoton = MessageBoxButtons.YesNoCancel;
    else if (sender == radioButton5)
        tipoDeBoton = MessageBoxButtons.RetryCancel;
    else
        tipoDeBoton = MessageBoxButtons.AbortRetryIgnore;
}
```

En el código anterior se investiga cual de los radioButtons (mediante sender) cambió su estatus de desactivado a activado y se pone la variable tipoDeBoton con uno de los 6 posibles valores que tiene la enumeración MessageBoxButtons.

Ahora programemos el evento de que se genere un cambio en la selección de los botones de la derecha es decir en Tipo de Icono, teniendo seleccionado el radioButton7 con etiqueta Information, hagamos clic en el botón que tiene un rayo en la ventana de propiedades (para seleccionar eventos) y seleccionemos CheckedChange, a la derecha escribamos como nombre para el método que se va a ejecutar cuando este evento se lleve a cabo, tipoDeIcono_CheckedChanged, hagamos lo mismo para los tres radioButtons restantes seleccionando siempre el mismo método.



Lo que sigue es agregar una variable privada a la clase y lo hacemos también antes del constructor:

```
private MessageBoxIcon tipoDeIcono = MessageBoxIcon.Error;
```

Luego insertemos el siguiente código para el método tipoDeIcono_CheckedChanged

```
private void tipoDeIcono_CheckedChanged(object sender, EventArgs e)
{
    if (sender == radioButton7) // display error icon
        tipoDeIcono = MessageBoxIcon.Information;
    else if (sender == radioButton8)
```

```

        tipoDeIcono = MessageBoxIcon.Exclamation;
    else if (sender == radioButton9)
        tipoDeIcono = MessageBoxIcon.Question;
    else // only one option left--display question mark
        tipoDeIcono = MessageBoxIcon.Error;
}

```

Como en el primer método, se investiga cual de los radioButtons (mediante sender) cambió su estatus de desactivado a activado y se pone la variable tipoDeIcono con uno de los cuatro posibles valores que tiene la enumeración MessageBoxIcon.

Por último queremos programar el botón Mostrar para que al hacer clic sobre él se ejecute el método button1_Click desplegando el MessageBox configurado con el valor que en su momento tengan las variables tipoDeBoton y tipoDeIcono.

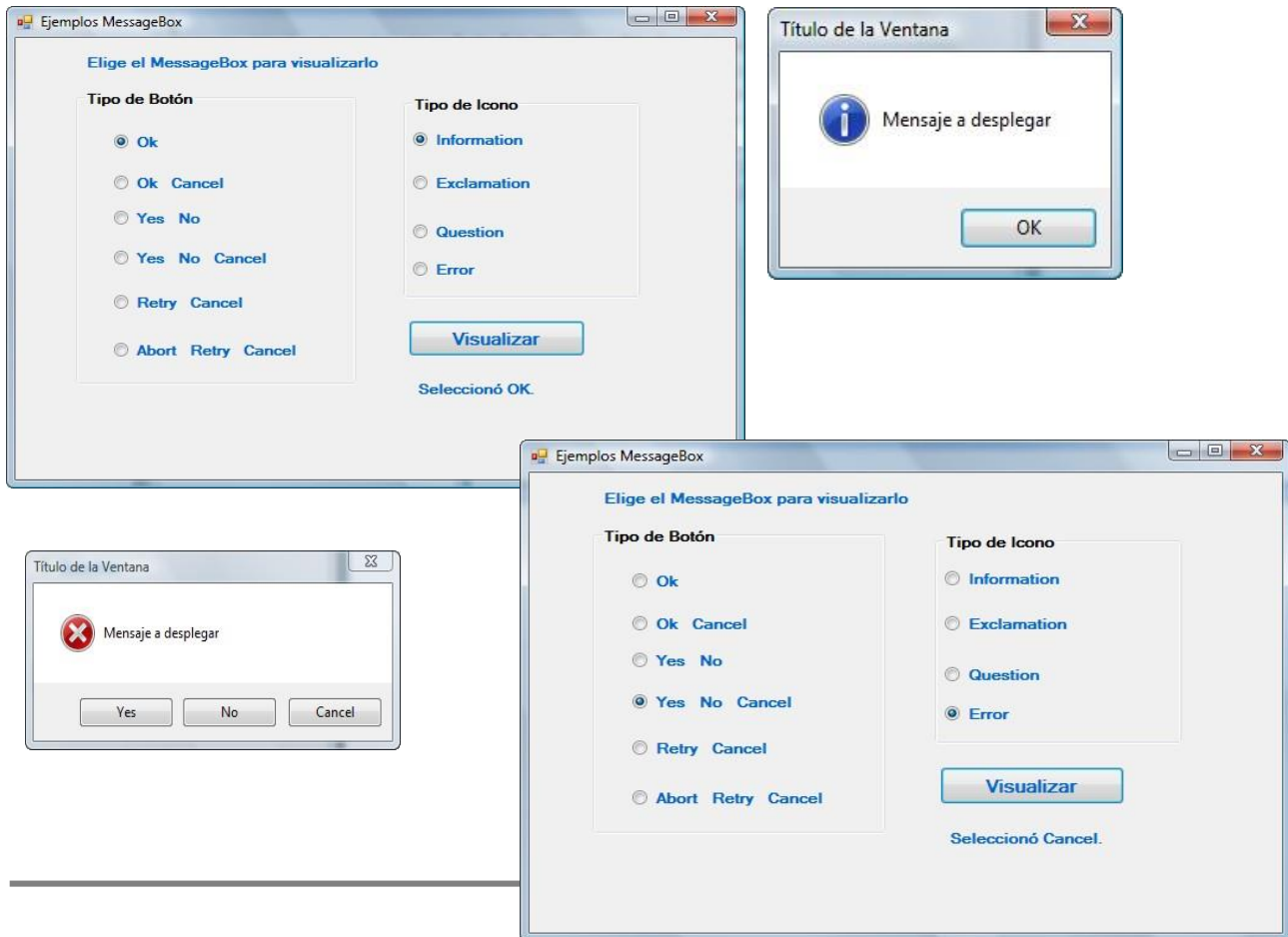
```

private void button1_Click(object sender, EventArgs e)
{
    DialogResult result =
    MessageBox.Show("Mensaje a desplegar",
        "Título de la Ventana", tipoDeBoton, tipoDeIcono);

    switch (result)
    {

        case DialogResult.OK: label2.Text = "Seleccionó OK."; break;
        case DialogResult.Cancel: label2.Text = "Seleccionó
Cancel."; break; case DialogResult.Yes: label2.Text =
"Seleccionó Yes."; break;
        case DialogResult.No: label2.Text = "Seleccionó No."; break;
        case DialogResult.Ignore: label2.Text = "Seleccionó
Ignore."; break; case DialogResult.Abort: label2.Text =
"Seleccionó Abort."; break; case DialogResult.Retry:
label2.Text = "Seleccionó Retry."; break;
    }
}

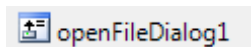
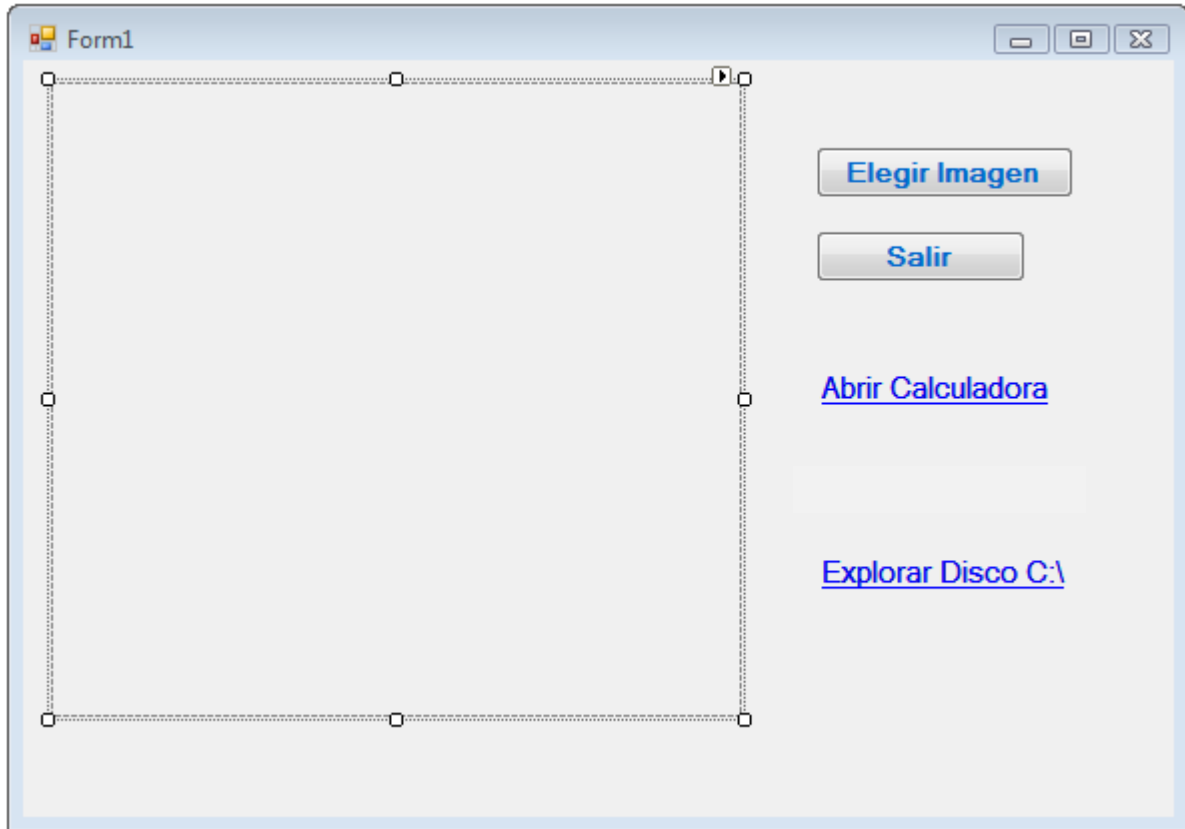
```



VISOR DE IMÁGENES Y ENLACES





Este proyecto abarca dos secciones, en la primera se trata de un visor de imágenes, es una aplicación muy simple que lee archivos gráficos, el botón Elegir Imagen en combinación con el control openFileDialog nos permite seleccionar la ubicación de nuestra imagen, una vez seleccionada la imagen se mostrara en el recuadro pictureBox1. La segunda parte son los enlaces a tres elementos el primero abre la calculadora de Windows, la segunda abre una liga a la página de la Universidad, por último abre el Explorador de Windows en el disco local C:\, esto se logra mediante los controles linkLabel1.

La siguiente figura muestra los controles necesarios para elaborar nuestro proyecto



Noten que el control openFileDialog1 no aparece dentro de la forma si no en la parte inferior de nuestro proyecto.,

Los controles necesarios para llevar a cabo el proyecto son:

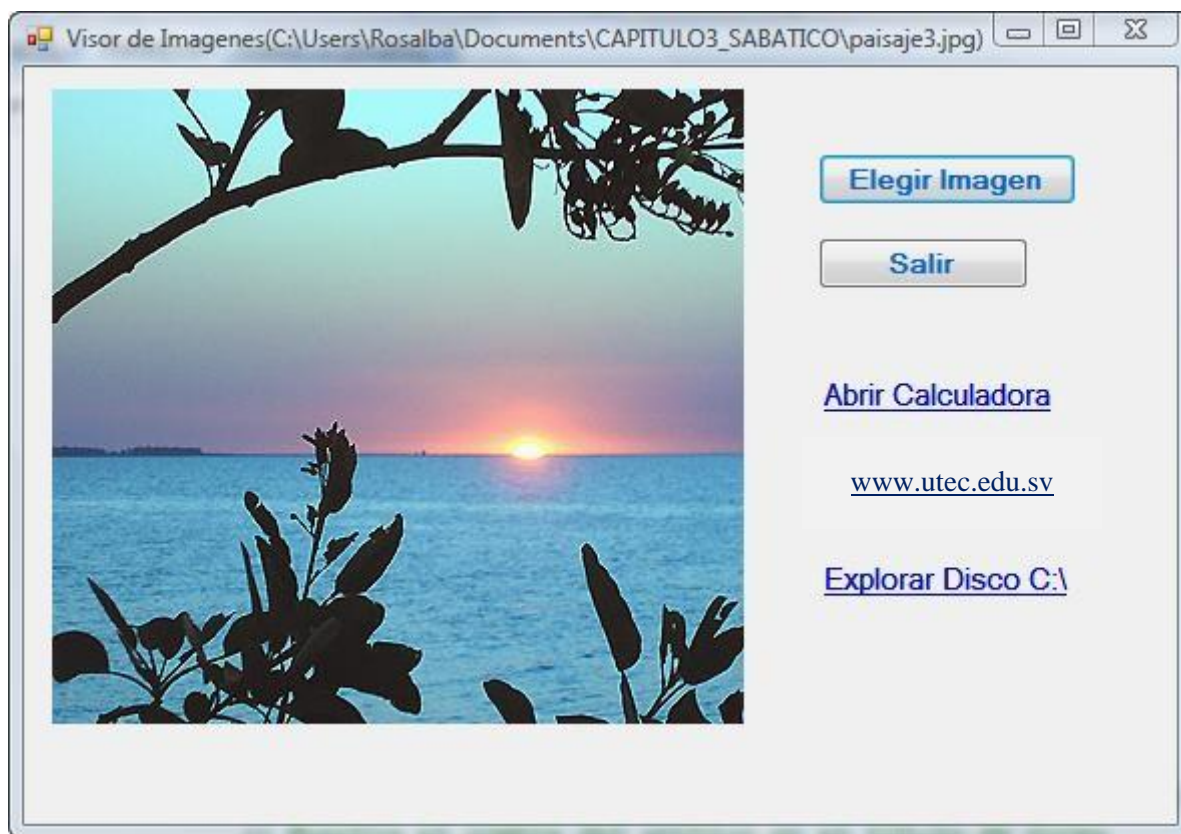
- 1 Caja de Imagen (pictureBox)  PictureBox
- 1 Caja de dialogo para abrir un archivo (openFileDialog)  OpenFileDialog
- 2 Botones (Button)  Button
- 3 Etiquetas de Enlace (linkLabel)  LinkLabel

Al botón 1 que dice Elegir Imagen al darle doble clic nos trasladar al código e insertamos lo siguiente:

```
private void button1_Click(object sender, EventArgs e)
{
    if (openFileDialog1.ShowDialog() == DialogResult.OK)
    { // Carga la imagen dentro del cuadro picture box.
        pictureBox1.Image= Image.FromFile(openFileDialog1.FileName);
        // Muestra el nombre del archivo en el titulo de forma

        this.Text = string.Concat("Visor de Imagenes(" + openFileDialog1.FileName
+ ")"); }
}
```

Un ejemplo de la ejecución del proyecto con esta primera parte del Visor de Imágenes:



En la segunda parte del proyecto se refiere a los Enlaces, usamos el control `linkLabel1` el cual tiene la peculiaridad de poseer una liga (como las páginas Web) a directorios de nuestro

sistema de archivos, a direcciones electrónicas de sitios Web o FTP y también a aplicaciones que se encuentren en el Registro de Windows.

Ya tenemos agregados los tres linkLabel1, ahora necesitamos la programación de los eventos de hacer click en el control, en los tres eventos utilizaremos la llamada del método:

```
System.Diagnostics.Process.Start(parametros);
```

En donde los parametros pueden ser:

- Una cadena de caracteres con el nombre de un directorio ejemplo. "C:\\Documents and Settings\\users\\orlandogiron"
- Una cadena de caracteres con el nombre de un programa Navegador de Internet y una segunda cadena con una dirección electrónica ejemplo. "http://google.com" que se quiere visitar
- Una cadena de caracteres con el nombre de un programa que se quiere mandar ejecutar
- En cada una de las etiquetas de enlace insertamos el siguiente código:

```
private void linkLabel1_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    linkLabel1.LinkVisited = true;
    System.Diagnostics.Process.Start("Calc");
}
```

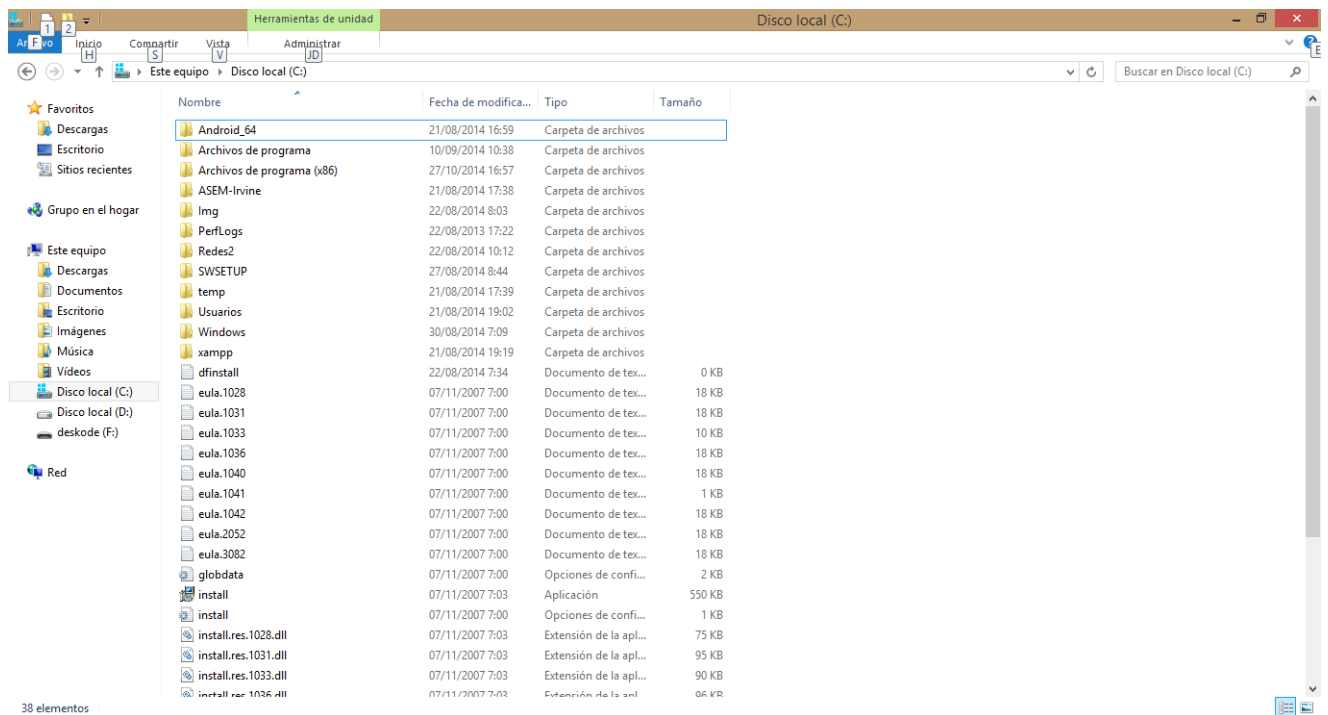
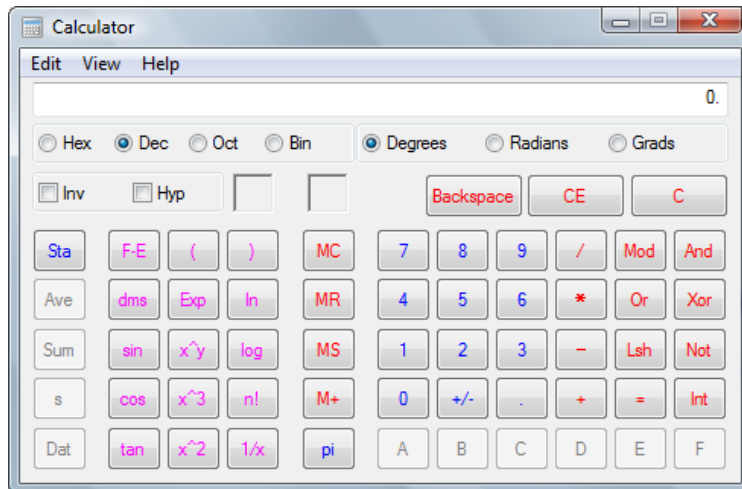
```
private void linkLabel2_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    linkLabel2.LinkVisited = true;
    System.Diagnostics.Process.Start("IExplore",
"http://www.utec.edu.sv");
}
```

```
private void linkLabel3_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    linkLabel3.LinkVisited = true;
    System.Diagnostics.Process.Start("C:\\");
}
```

Observemos que normalmente no cambiamos el nombre de los controles por ejemplo aquí utilizamos el nombre por default linkLabel1, linkLabel2, linkLabel3, esto digamos que

es por “simplicidad”, aunque si quisiéramos que fuera mas intuitivo nuestro programa podríamos llamar en lugar de linkLabel11, quizás calculadora, o en lugar de linkLabel12 el nombre de uacm, etc.

La ejecución de esta segunda parte del proyecto abrirá las siguientes aplicaciones:



Universidad *Tecnológica*
de El Salvador



La Universidad ▼

Nuevo Ingreso ▼

Facultades ▼

Vicerrectoría de Investigación

Vicerrectoría de Proyección Social



UN PAÍS **PROGRESA** CUANDO
TÚ PROGRESAS

INSCRÍBETE HOY
EN EL CICLO
01-2015
Y RESERVA TUS HORARIOS

LICENCIATURAS
INGENIERÍAS
ARQUITECTURA
TÉCNICOS
MAESTRÍAS

PRESENCIALES
BILINGÜES
VIRTUALES

LO QUE CUENTA ES TU **ACTITUD POSITIVA**



Portal Educativo

Consulta de notas,
correo, inscripción de
materias y más.
Solo alumnos activos



Publicaciones

Noticias y eventos

2 Lanzamiento del quinto número de la
revista
22-10-2014



Bolsa de



Graduados



Biblioteca



EDUTEC



MUA



Videos



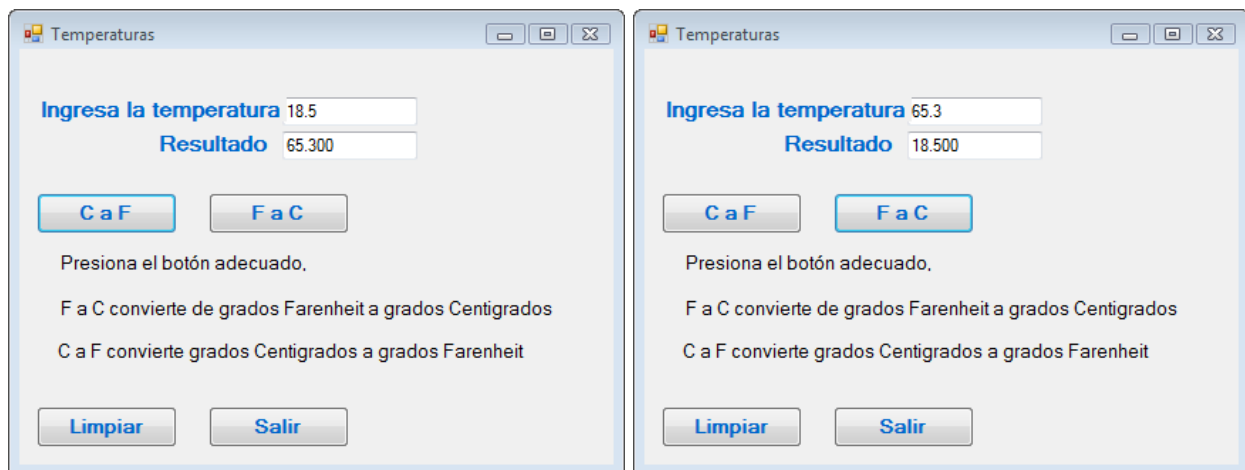
Radios Utec
on line

CALCULO DE FAHRENHEIT A CENTIGRADOS

Un segundo ejercicio, crearemos un convertidor de temperaturas de grados Centígrados a grados Fahrenheit, y viceversa, el usuario tiene la opción de elegir en qué sentido desea realizar la conversión, la siguiente figura muestra los controles insertados en la forma.

Ejemplo de la ejecución:

Del lado izquierdo convierte de C a F y del lado derecho convierte de F a C



El código insertado para convertir de C a F es:

```
private void button1_Click(object sender, EventArgs e)
{
    double gfahrenheit, ct1;
    ct1 = Convert.ToDouble(textBox1.Text);
    gfahrenheit = ct1 * 1.8 + 32.0;
}
```

El código insertado para convertir de F a C es:

```
private void button2_Click(object sender, EventArgs e)
{
    double gcentigrados, ct1;
    ct1 = Convert.ToDouble(textBox1.Text);
    gcentigrados = (ct1 - 32.0)/1.8;

}
```

El código para los botones de Limpiar y Salir, son idénticos al del ejemplo Multiplica.