Controles

Timer

ImageList

pictureBox

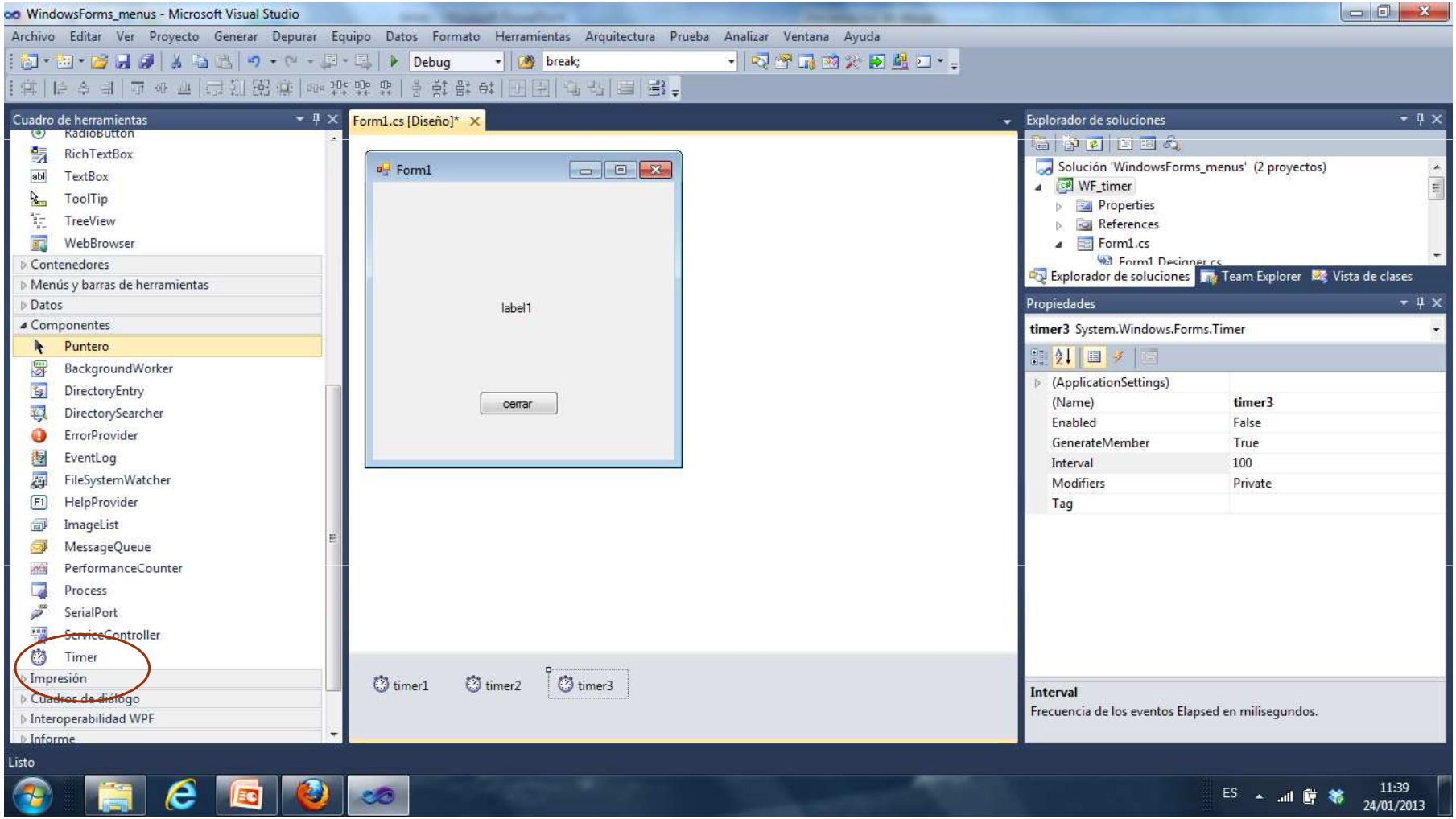
progressBar

Timer

- Si se desea que una acción suceda con cierta periodicidad se puede utilizar un control ***Timer.***
- Este control produce de modo automático un evento cada cierto número de milisegundos y es de fundamental importancia para crear ***animaciones o aplicaciones con movimiento de objetos. La*** propiedad más importante de un objeto de este tipo es ***Interval, que determina, precisamente, el*** intervalo en milisegundos entre eventos consecutivos. La acción que se desea activar debe programarse en el evento ***Tick de ese mismo control.***
- Si en algún momento se desea detener momentáneamente la acción periódica es suficiente con hacer ***False la propiedad Enabled del control Timer y para arrancarla de nuevo volver a hacer True*** esa propiedad.
- Haciendo 0 la propiedad ***Interval también se consigue inhabilitar el Timer.***

Ejemplo

- Crear un proyecto y añadir al formulario una etiqueta.
  - Las propiedades de la etiqueta serán:
    - Autosize=False
    - TextAlign=MiddleCenter
- Agregamos a nuestro formulario un botón y le cambiamos el nombre a “cerrar”, este botón trabajará con los timers
- Y agregamos 3 timers


WindowsForms_menus - Microsoft Visual Studio
Archivo Editar Ver Proyecto Generar Depurar Equipo Datos Formato Herramientas Arquitectura Prueba Analizar Ventana Ayuda
Debug
break;
Cuadro de herramientas
RadioButton
RichTextBox
TextBox
ToolTip
TreeView
WebBrowser
Contenedores
Menús y barras de herramientas
Datos
Componentes
Puntero
BackgroundWorker
DirectoryEntry
DirectorySearcher
ErrorProvider
EventLog
FileSystemWatcher
HelpProvider
ImageList
MessageQueue
PerformanceCounter
Process
SerialPort
ServiceController
Timer
Impresión
Cuadros de diálogo
Interoperabilidad WPF
Informe
Form1.cs [Diseño]*
Form1
label1
cerrar
timer1
timer2
timer3
Explorador de soluciones
Solución 'WindowsForms_menus' (2 proyectos)
WF_timer
Properties
References
Form1.cs
Explorador de soluciones
Team Explorer
Vista de clases
Propiedades
timer3 System.Windows.Forms.Timer
(ApplicationSettings)
(Name) timer3
Enabled False
GenerateMember True
Interval 100
Modifiers Private
Tag
Interval
Frecuencia de los eventos Elapsed en milisegundos.
Listo
ES
11:39
24/01/2013

- Vamos a las propiedades de los timers y le cambiamos los intervalos
- Al timer1 le ponemos 100 milisegundos y al 2 y al 3 en 5000 milisegundos
- Ahora vamos a arrancar el temporizador en el evento_click del botón

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Start();
}
```

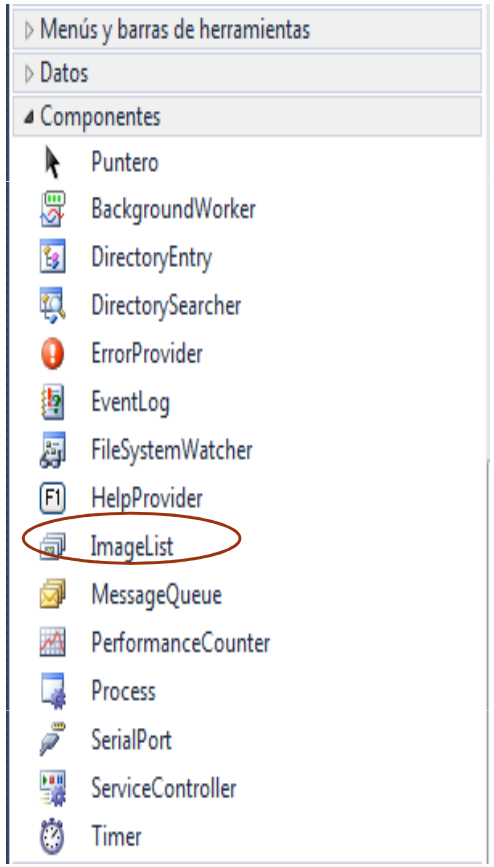
- Ahora vamos a crear el código en el evento del timer, haciedo doble click en el control timer1 y sucesivos comola figura siguiente….

```
private void button1_Click(object sender, EventArgs e)
{
    timer1.Start(); //Inciamos el timer1
}

private void timer1_Tick(object sender, EventArgs e)
{
    label1.Text = "cerrando sesion"; //cambia la etiqueta de nombre
    timer1.Stop(); // para el timer1
    timer2.Start(); //inicia el timer2

}

private void timer2_Tick(object sender, EventArgs e)
{
    label1.Text = "cerrando la aplicación"; //cambia la etiqueta de nombre
    timer2.Stop();//lo para...
    timer3.Start();//e inicia el 3
}

private void timer3_Tick(object sender, EventArgs e)
{
    label1.Visible = false; //esconde la etieuta
     Close();
}
```

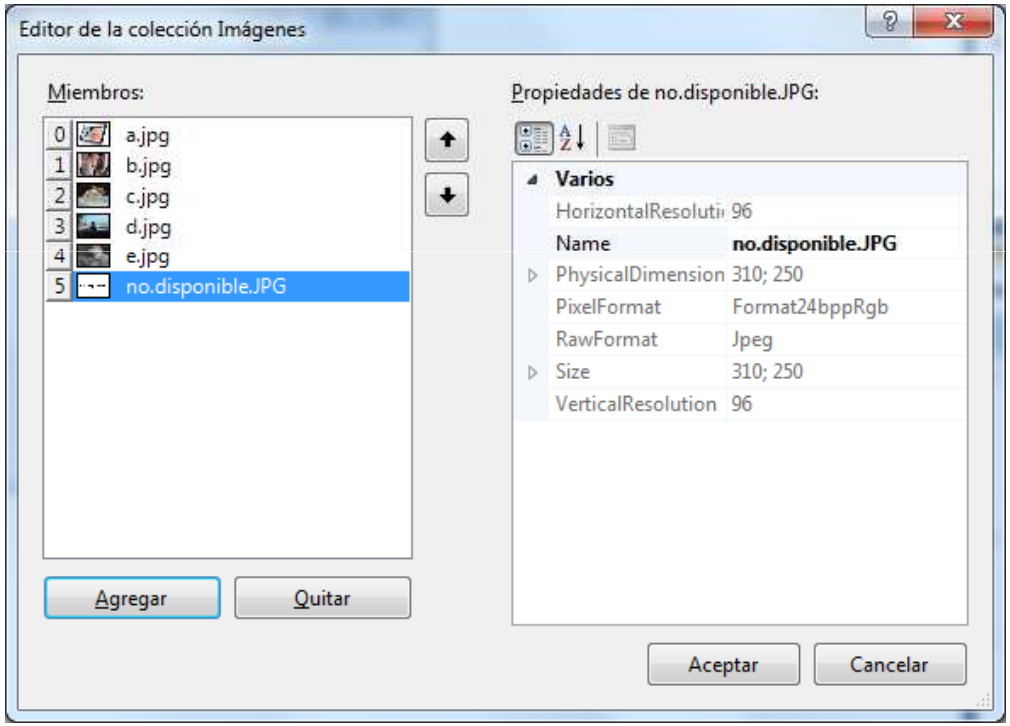
¿… y que vamos a conseguir con todo esto?

Imaginar que al finalizar la sesión la aplicación tiene que guardar una serie de cambios… si no se le avisa al usuario de alguna manera puede pensar que se la bloqueado…

ImageList

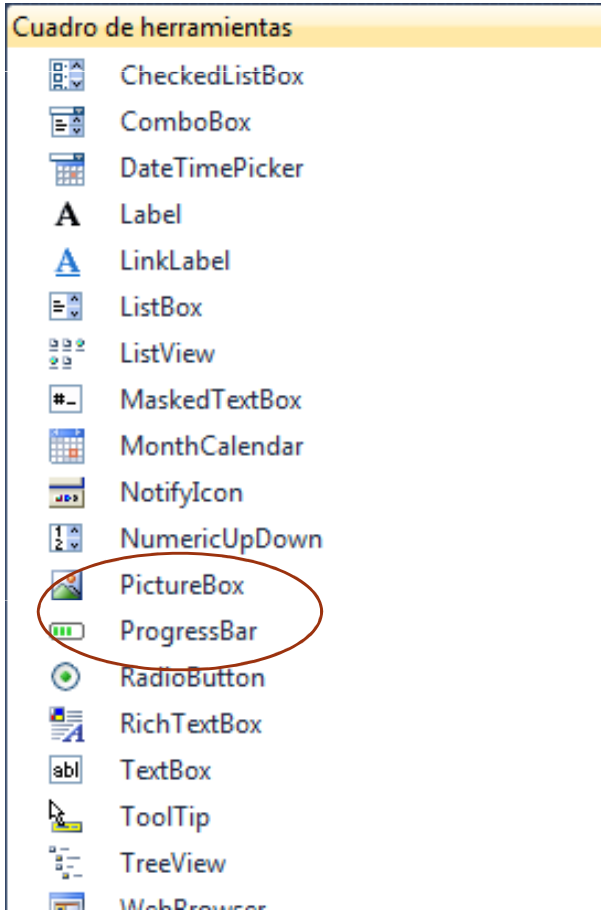


- Este control actúa como repositorio de imágenes, del que se alimentarán otros controles del formulario que necesiten mostrar gráficos en su interior.
- Una vez añadido este control en el formulario y haciendo clic en su propiedad Images, se abrirá una ventana en la que podremos añadir y quitar las imágenes que van a formar parte de la lista del control, así como ver en el panel complementario, la información sobre cada imagen asignada:



PictureBox

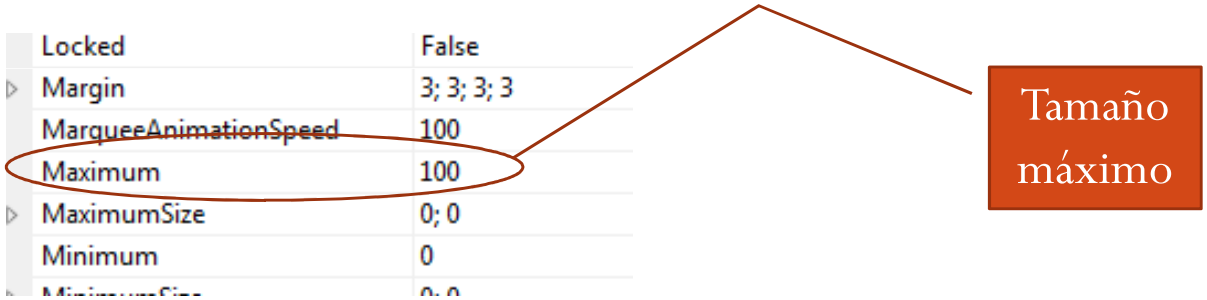
- Muestra una imagen que proviene de un imagelist




```
pictureBox1.Image = imageList1.Images[i];
```

progressBar

- propiedades



progressBar1.Value = 0; **Inicializar…**

progressBar1.Increment(+1); **Incrementar…**