Programación con C# .NET

Menús y Barras de Herramientas

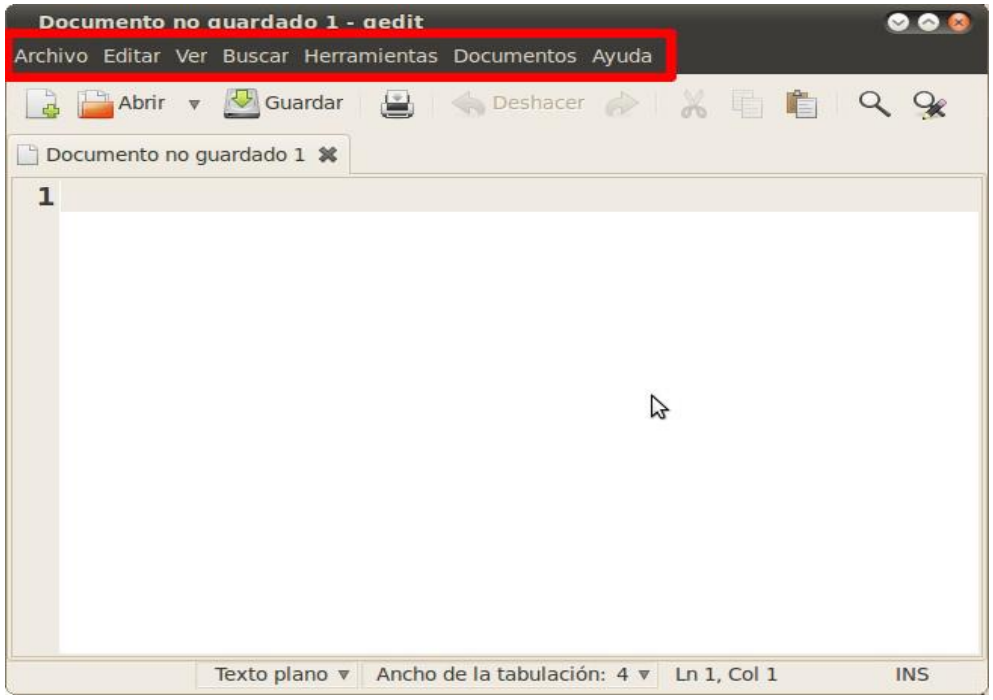
Contenido

- Barra de menús
- Clases para trabajar con menús
- Diseño de una barra de menús
- Crear un menú mediante programación
- Controlador de un elemento de un menú
- Nemónicos
- Aceleradores
- Barra de herramientas
- Barra de estado
- Menús dinámicos
- Bibliografía

Microsoft .net

Barra de Menús

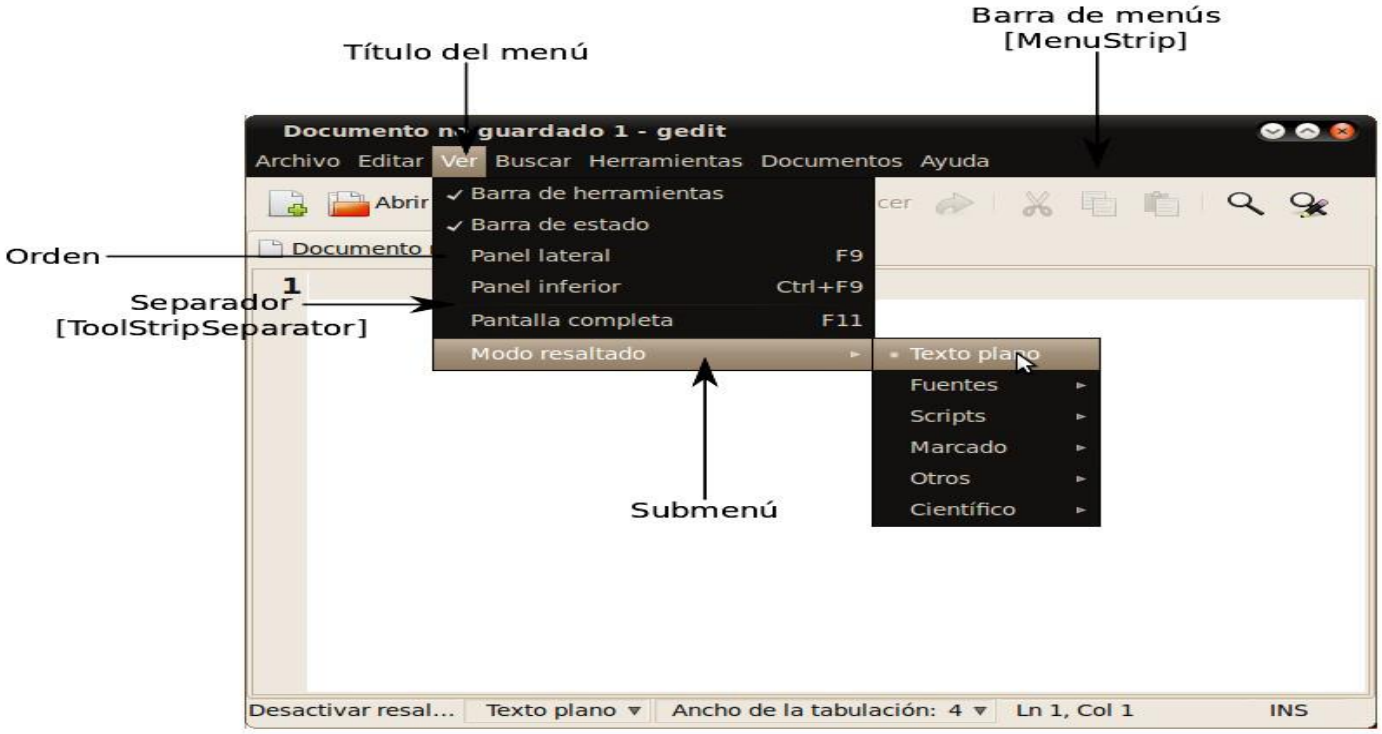
- Un menú es una forma de proveer al usuario un conjunto de órdenes, lógicamente relacionadas, agrupadas bajo un mismo título.
- Los elementos de un menú pueden ser órdenes, submenús y separadores
  - Al dar clic sobre una orden de un menú se ejecuta una acción o se despliega una caja de diálogo.
  - Una orden seguida de tres puntos (...) indica que se desplegará una caja de diálogo.
  - Un separador tiene como finalidad separar grupos de órdenes de un mismo menú, lógicamente en función de su actividad.



Microsoft .net

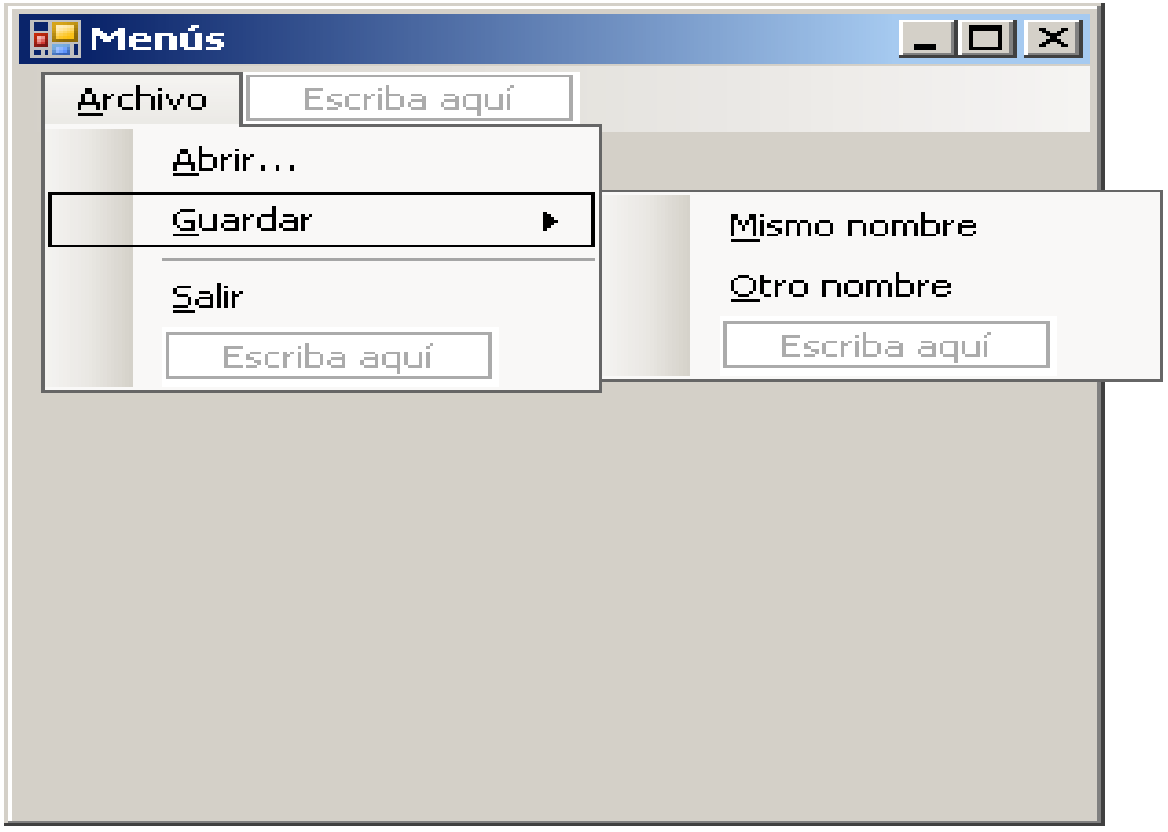
Clases para trabajar con menús

- Las clases que permiten trabajar con las barra de menús son:
  - **MenuStrip** Representa una barra de menús
  - **ToolStripMenuItem** Representa los elementos de los menús
  - **ToolStripSeparator** Representa un separador



Diseño de una barra de menús

- Para diseñar un menú, utilizaremos el *editor de menús*
  - Basta con rellenar las cajas de texto *"Escriba aquí" con los nombres de* menús, submenús, órdenes o separadores que sean necesarios.



Microsoft .net

Diseño de una barra de menús

- Para crear una barra de menús, los pasos son:
  - Arrastrar desde la caja de herramientas un control del tipo **MenuStrip** sobre el formulario.
  - Introducir el título del menú.
    - Escriba en la caja de texto "*Escriba aquí" el título del menú que* desea crear.
    - Inserte un ampersand (&) antes de la letra que quiere que de acceso directo al menú.
  - Introducir los elementos que componen el menú
  - Crear un submenú
  - Añadir un separador
    - Escribir un guión (-) en la caja de texto "*Escriba aquí"*
  - Cerrar el editor de menús
    - Haga clic en cualquier lugar fuera de los menús

Microsoft .net

Crear un menú por programación

Crear la barra de menús

- El código generado por el asistente para crear la barra de menús es el siguiente:

```
private System.Windows.Forms.MenuStrip menuStrip1;
//...
this.menuStrip1 = new System.Windows.Forms.MenuStrip();
this.menuStrip1.Location = new System.Drawing.Point(0, 0);
this.menuStrip1.Name = "menuStrip1";
this.menuStrip1.Size = new System.Drawing.Size(292, 24);
//...
this.Controls.Add(menuStrip1);
```

Crear un menú por programación

Crear un menú

- El Para crear un nuevo menú, hay que construir un objeto de la clase **ToolStripMenuItem, establecer sus propiedades y añadirlo a la barra de** Menús:

```
private System.Windows.Forms.ToolStripMenuItem archivoToolStripMenuItem;
//…
this.archivoToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.archivoToolStripMenuItem.Name = "archivoToolStripMenuItem";
this.archivoToolStripMenuItem.Text = "&Archivo";
this.menuStrip1.Items.AddRange(new ToolStripItem[] {archivoToolStripMenuItem});
```

- *AddRange añade uno o más menús a la colección de objetos ToolStripItem* referenciada por la propiedad *Items del control barra de menús.*

Microsoft .net

Crear un menú por programación

Crear un elemento de un menú

- Para crear un elemento de un menú, por ejemplo *Archivo-Abrir, hay que* construir un objeto de la clase ***ToolStripMenuItem****, establecer sus* propiedades y añadirlo al menú:

```
private System.Windows.Forms.ToolStripMenuItem abrirToolStripMenuItem;
//...
this.abrirToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.abrirToolStripMenuItem.Name = "abrirToolStripMenuItem";
this.abrirToolStripMenuItem.Text = "&Abrir...";
this.archivoToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {abrirToolStripMenuItem});
```

- Se utiliza ***AddRange*** *para añadir un elemento a la colección de elementos* del contenedor asociado con el menú Archivo
  - Dicha colección está referenciada por la propiedad ***DropDownItems*** *del* menú

Microsoft .net

Crear un menú por programación

Crear un submenú

- Un elemento puede ser también un submenú
  - Se trata de un objeto *ToolStripMenuItem añadido al menú*
  - El código siguiente crea el submenú *Archivo-Guardar y le añade dos elementos ToolStripMenuItem que supondremos creados.*

```
private System.Windows.Forms.ToolStripMenuItem guardarToolStripMenuItem;
//…
this.guardarToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.guardarToolStripMenuItem.Name = "guardarToolStripMenuItem";
this.guardarToolStripMenuItem.Text = "&Guardar";
//…
this.guardarToolStripMenuItem.DropDownItems.AddRange(newSystem.Windows.Forms.To
olStripItem[] {this.mismoNombreToolStripMenuItem,
this.otroNombreToolStripMenuItem});
//…
this.archivoToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {this.guardarToolStripMenuItem});
```

Microsoft .net

Crear un menú por programación

Crear un separador

- Los separadores se utilizan para agrupar las órdenes en función de su actividad.
- Un separador es un objeto de la clase *ToolStripSeparator.*
- El siguiente ejemplo añade un separador al menú *Archivo antes de la orden Salir.*

```
private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
//...
this.toolStripSeparator1 = new
System.Windows.Forms.ToolStripSeparator();
this.toolStripSeparator1.Name = "toolStripSeparator1";
//...
this.archivoToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {this.toolStripSeparator1});
```

- Los menús se añaden a la barra de menús en el orden en el que se ejecuten las llamadas al método AddRange.

Microsoft .net

Controlador de un elemento de un menú

- Para escribir un método que responda al mensaje que se genera cuando un usuario hace clic en una orden de un menú, hay que asignar a dicha orden un controlador de eventos *Click.*
- Por ejemplo, para asociar un controlador de eventos *Click con la orden Salir del menú Archivo, el código es el siguiente:*

```
ArchivoSalir.Click += new EventHandler(ArchivoSalir_Click);

private void ArchivoSalir_Click(object sender, EventArgs e)
{
        Close();
}
```

Microsoft .net

Nemonicos

- El nemónico de un menú o de alguno de sus elementos proporciona acceso directo desde el teclado.
- Un nemónico se distingue porque se corresponde con el carácter que aparece subrayado en el texto del control.
- El acceso desde el teclado se consigue pulsando las teclas **Alt + nemónico**

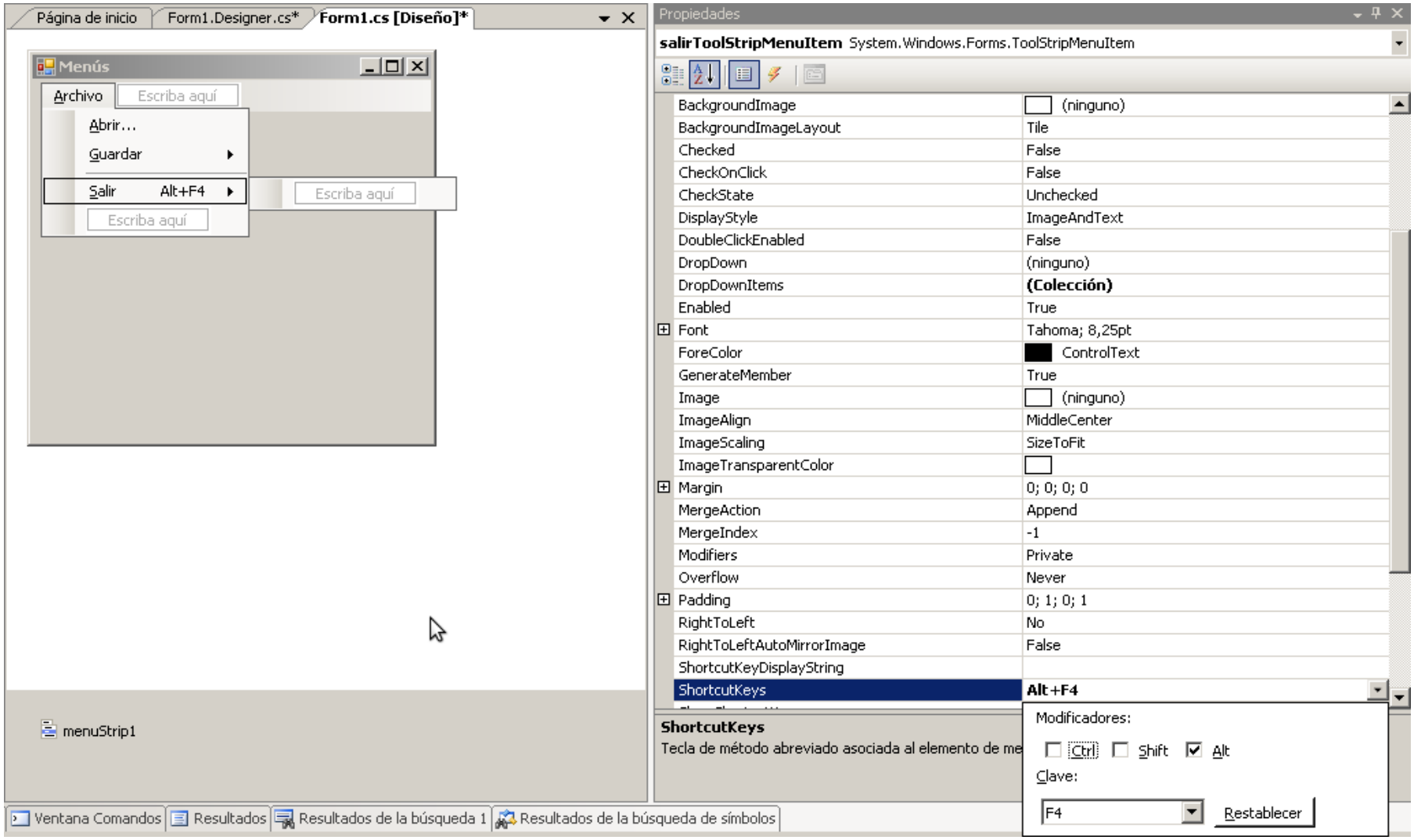
Microsoft .net

Aceleradores

- Los aceleradores son muy similares a los nemónicos, excepto en que se puede especificar cualquier combinación de teclas de la forma:
  - **[{Ctrl | Shift | Alt}] + {cualquier tecla}**
- Un nemónico actúa independientemente de que el componente este o no siendo mostrado.
- Los aceleradores sólo pueden estar asociados con elementos de un menú, no con menús.
- La propiedad *ShortCutKeys de la clase* ***ToolStripMenuItem*** *permite* establecer un acelerador para un elemento de un menú.
- Tiene un parámetro que se corresponde con un valor de tipo enumerado *Keys.*
- Este valor especificará la tecla o combinación de teclas que darán lugar al acelerador.

Microsoft .net

Aceleradores



Microsoft .net

Barra de herramientas

- Es una ventana que normalmente contiene botones gráficos, aunque también puede contener otros controles, como por ejemplo, cajas de texto o listas desplegables.
- Un clic en un botón de la barra de herramientas produce los mismos eventos que un clic en la orden de un menú que realice su misma función.
- Normalmente se añadirá un botón a una barra de herramientas para evitar al usuario la molestia de abrir un menú para ejecutar una determinada orden.
- Una barra de herramientas es un objeto **ToolStrip** que actúa como contenedor para los objetos de las clases **ToolStripButton, ToolStripLabel, ToolStripDropDownButton, ToolStripSeparator, ToolStripComboBox, ToolStripTextBox, etc.**

Microsoft .net

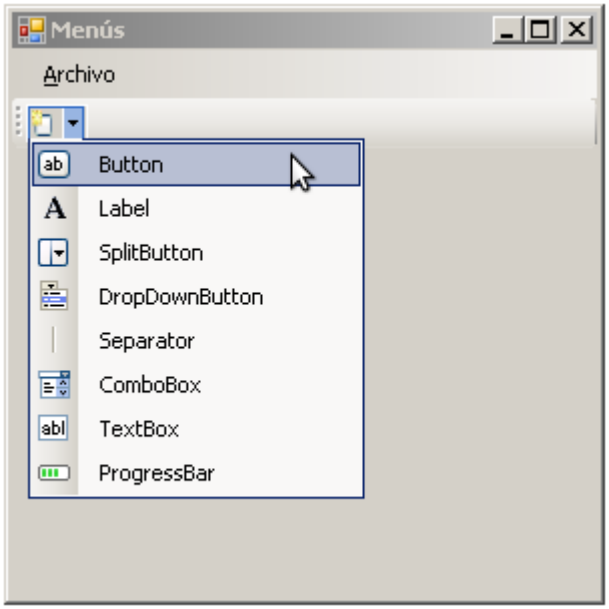
Barra de herramientas

- El siguiente código se genera cuando añadimos al formulario un control **ToolStrip** desde la caja de herramientas:

```
private System.Windows.Forms.ToolStrip barraHerramientas;
//...
this.barraHerramientas = new
System.Windows.Forms.ToolStrip();
this.barraHerramientas.Name = "barraHerramientas";
this.barraHerramientas.Size = new System.Drawing.Size(292, 25);
//...
this.Controls.Add(this.barraHerramientas);
```

Barra de herramientas

- Para añadir los botones a la barra de herramientas, podemos hacer uso de un botón que se encuentra en la barra para tal fin.
- Podemos escoger el tipo de elemento que se añadirá a la barra.
- Cada botón es un objeto de la clase **ToolStripButton** con una imagen predeterminada.
- Podemos cambiar la imagen predeterminada, si así lo deseamos.



Microsoft .net

Barra de herramientas

- Una vez realizado el proceso, el asistente habrá añadido, por ejemplo, a la clase el siguiente código:

```
private System.Windows.Forms.ToolStripButton btAbrir;
this.btAbrir = new System.Windows.Forms.ToolStripButton();
this.btAbrir.Name = "btAbrir";
this.btAbrir.Image = ((System.Drawing.Image)(resources.GetObject("btAbrir.Image")));
this.btAbrir.ImageTransparentColor = System.Drawing.Color.Magenta;
this.barraHerramientas.Items.AddRange(new System.Windows.Forms.ToolStripItem[]
{this.btAbrir});
```

- Ahora, hay que asociar con cada botón el mismo controlador de eventos que tienen asociados sus órdenes respectivas.
  - Para ello, desde la lista de los eventos hay que elegir el evento *Click y* asociarlo con el controlador adecuado.
  - En el caso del ejemplo, el mismo controlador de la orden *Abrir del* menú *Archivo (ArchivoAbrir_Click).*

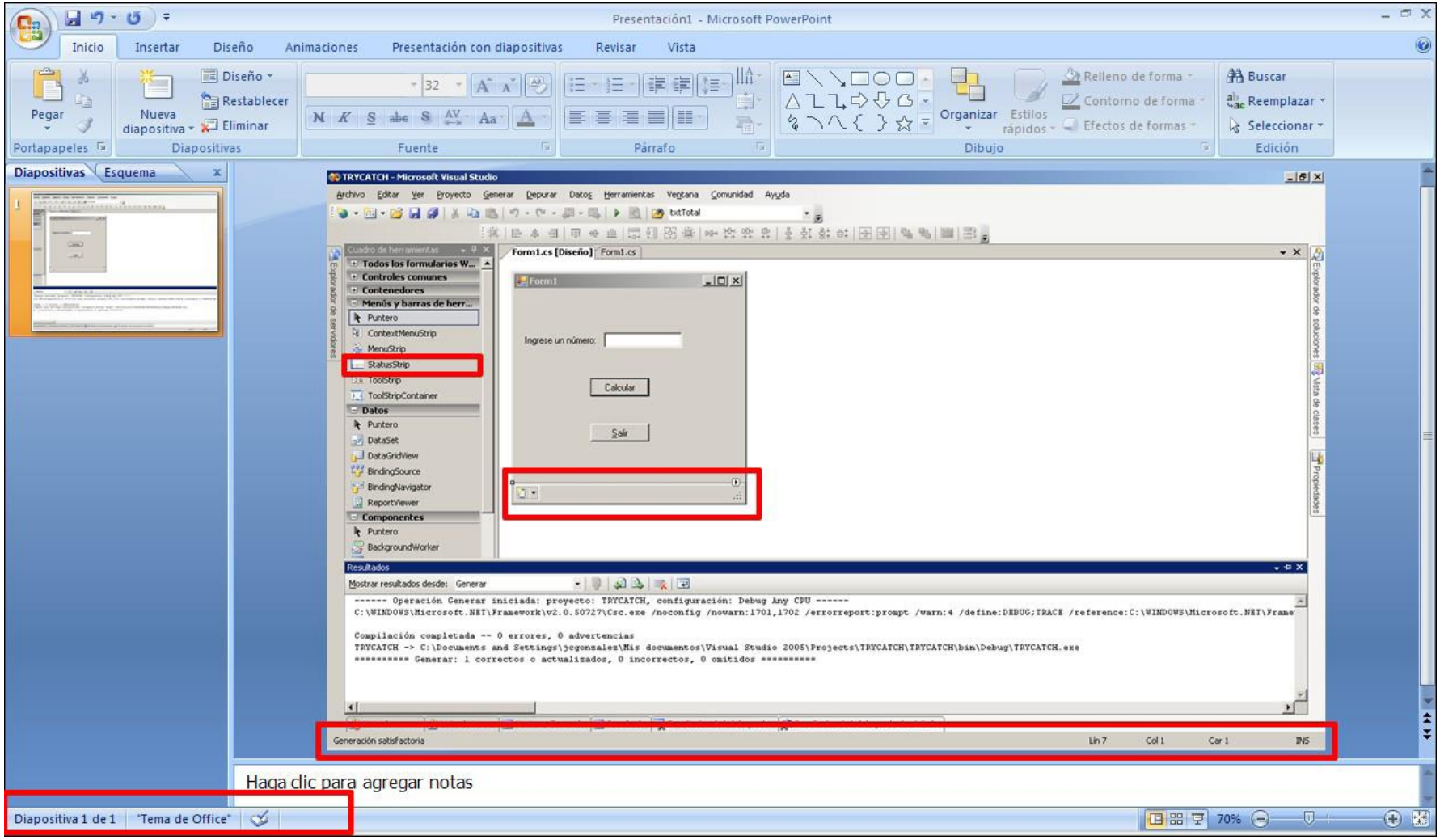
Microsoft .net

Barra de estado

- Control que se visualiza normalmente en la parte inferior de una ventana.
- Contiene paneles (etiquetas) que muestran diferentes tipos de Información.
  - Puede contener otros controles, por ejemplo, botones o barras de Progreso.
- Una barra de estado es un objeto de la clase **StatusStrip.**
  - Actúa como contenedor para los objetos de las clases: **ToolStripStatusLabel, ToolStripProgressBar, ToolStripDropDownButton y ToolStripSplitButton.**

Microsoft .net

Barra de estado



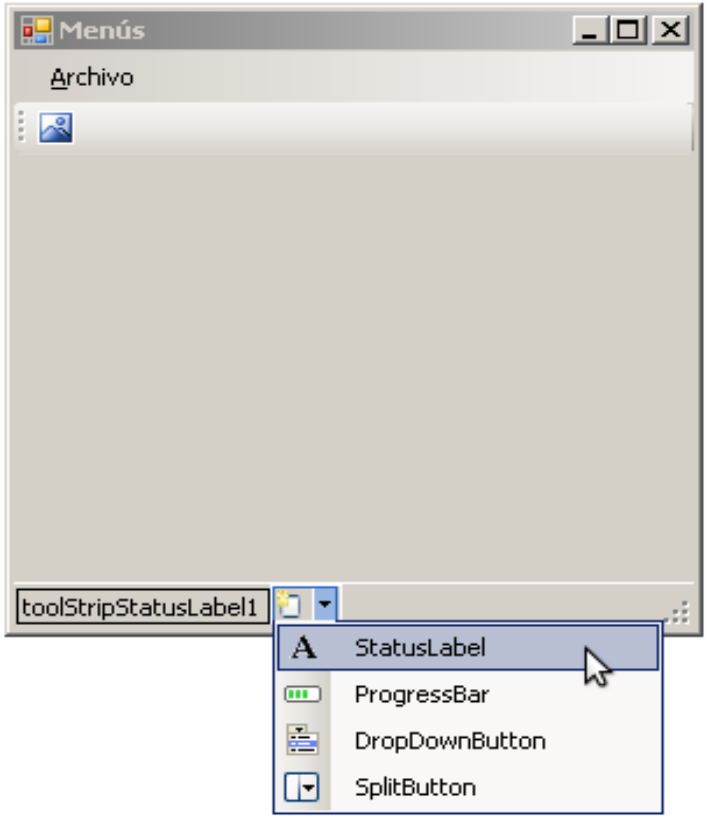
Barra de estado

- El código generado cuando añadimos una barra de estado, arrastrando desde la caja de herramientas un control **StatusStrip, sobre el formulario** es el siguiente:

```
private System.Windows.Forms.StatusStrip statusStrip1;
//...
this.statusStrip1 = new System.Windows.Forms.StatusStrip();
this.statusStrip1.Location = new System.Drawing.Point(0, 251);
this.statusStrip1.Name = "statusStrip1";
//...
this.Controls.Add(this.statusStrip1);
```

Barra de estado

- Para añadir un control a la barra de estado, ella misma proporciona un botón que permite esta acción.



Microsoft .net

Barra de estado

- El siguiente código muestra cómo se añade un control de la clase **ToolStripStatusLabel**

```
private System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel1;
//...
this.toolStripStatusLabel1 = new System.Windows.Forms.ToolStripStatusLabel();
this.toolStripStatusLabel1.Name = "toolStripStatusLabel1";
this.toolStripStatusLabel1.Text = "Listo";
//...
this.statusStrip1.Items.AddRange(new System.Windows.Forms.ToolStripItem[]
{this.toolStripStatusLabel1});
```

Barra de estado

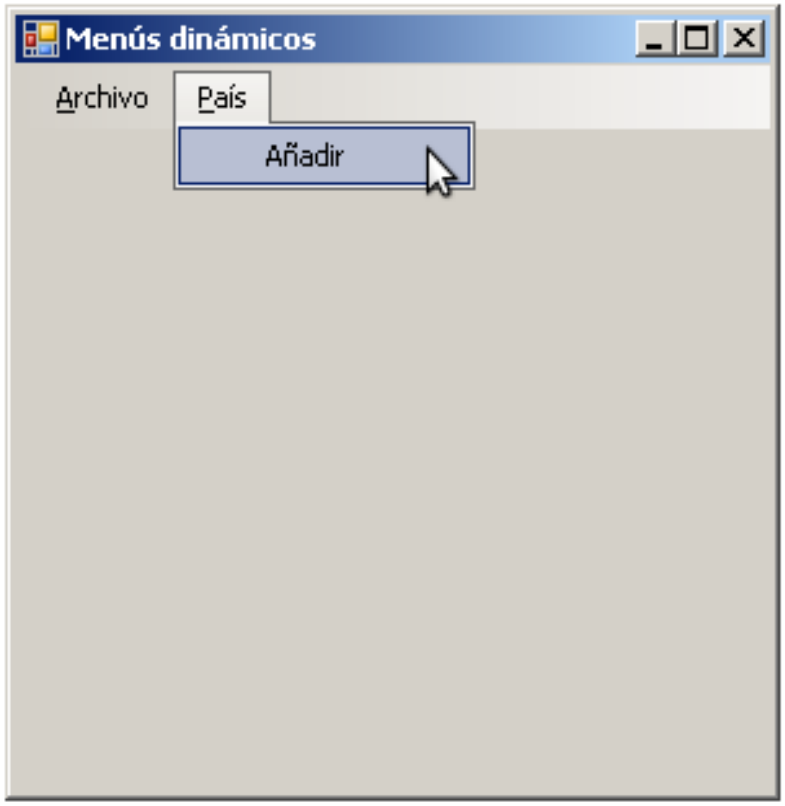
- Inicialmente la etiqueta *toolStripStatusLabel1 muestra el mensaje "Listo".*
- Para que muestre información acerca de la función que realiza la orden seleccionada de un menú, habrá que controlar cuándo el ratón entra en el elemento del menú (evento **MouseEnter**) y cuándo sale (evento **mouseLeave**)
- Ejemplo:

```
private void ArchivoSalir_MouseEnter(object sender, EventArgs e)
{
        toolStripStatusLabel1.Text = "Cierra la aplicación";
}
private void ArchivoSalir_MouseLeave(object sender, EventArgs e)
{
        toolStripStatusLabel1.Text = "Listo";
}
```

Microsoft .net

Menús Dinámicos

- Supóngase que se tiene el siguiente formulario:



Microsoft .net

Menús Dinámicos

- El controlador del evento *Click del elemento "Añadir".*
  - Nos permitirá añadir dinámicamente elementos al menú "*País".*
  - Los títulos de los elementos se corresponden con nombres de países (de la forma "*País #").*

```
private void PaisAnadir_Click(object sender, EventArgs e)
{
    // Construir un título para el menú
    string titulo;
    titulo = "País " + (menuPais.DropDownItems.Count);
    // Crear un elemento con título construido
    ToolStripMenuItem elemento = new ToolStripMenuItem(titulo);
    // Especificar cuál será su controlador de eventos Click
    elemento.Click += new EventHandler(ElementoMenuPais_Click);
    // Añadir el elemento al menú País
    menuPais.DropDownItems.Add(elemento);
}
```

Microsoft .net

Menús Dinámicos

- Ahora habrá que escribir el controlador del evento Click de los elementos añadidos dinámicamente.
- Para eso utilizaremos el método *ElementoMenuPais_Click*

```
private void ElementoMenuPais_Click(object sender, EventArgs e)
{
    //Obtenemos la referencia al elemento seleccionado
    ToolStripMenuItem elemento = (ToolStripMenuItem)sender;
    MessageBox.Show(elemento.Text);
}
```

- Lo anterior hace que al dar clic sobre un elemento añadido dinámicamente, se muestre el título de ese elemento por pantalla

Microsoft .net

Menús Dinámicos

- El resultado de ejecutar el código anterior sería:

Microsoft .net

Menús Dinámicos

- Ahora, vamos a añadir otra orden *"Eliminar" que permita eliminar el* último elemento añadido.
- También añadiremos un separador.
- Ahora, el menú "País" tendrá inicialmente tres elementos:
  - "Añadir"
  - "Eliminar"
  - Y el separador
- Primero añadimos los elementos *"Eliminar" (PaisEliminar) y el separador* (*Separador1) al menú "País".*
- Luego, asignamos a su propiedad *Visible el valor false.*
- Por último, modificamos el controlador *PaisAnadir_Click*

Microsoft .net

Menús Dinámicos

```
private void PaisAnadir_Click(object sender, EventArgs e)
{
    // Inicialmente hay 3 elementos (dos ocultos)
    if (menuPais.DropDownItems.Count == 3)
    {
        // Hacer visibles los elementos Eliminar y el Separador1
        PaisEliminar.Visible = true;
        Separador1.Visible = true;
    }
    // Construir un título para el menú
    string titulo;
    titulo = "País " + (menuPais.DropDownItems.Count - 2);
    // Crear un elemento con título construido
    ToolStripMenuItem elemento = new ToolStripMenuItem(titulo);
    // Especificar cuál será su controlador de eventos Click
    elemento.Click += new EventHandler(ElementoMenuPais_Click);
    // Añadir el elemento al menú País
    menuPais.DropDownItems.Add(elemento);
}
```

Microsoft .net

Menús Dinámicos

- Ahora falta colocar el código del evento *Click para la orden "Eliminar".*

```
private void PaisEliminar_Click(object sender, EventArgs e)
{
    // Eliminar el último elemento
    int indUltimo = menuPais.DropDownItems.Count - 1;
    menuPais.DropDownItems.RemoveAt(indUltimo);
    if (menuPais.DropDownItems.Count == 3)
    {
        // No quedan más países. Ocultar los elementos
        // Eliminar y el separador
        PaisEliminar.Visible = false;
        Separador1.Visible = false;
    }
}
```

- El método *RemoveAt permite eliminar el elemento de la colección que está en la* posición especificada (la primera posición es la cero)

Microsoft .net

Menús Emergentes

- **Tarea:**
  - Agregue al ejemplo anterior un menú emergente cuyo comportamiento será igual al menú Pais; con sus elementos añadir, eliminar y el separador.

Bibliografía

Enciclopedia de Microsoft Visual C#, 4ta edición

Fco. Javier Ceballos Sierra

RA-MA

Microsoft .net