



**UNIVERSIDAD TECNOLÓGICA NACIONAL**

***FACULTAD REGIONAL GENERAL PACHECO***

**TÉCNICO SUPERIOR EN PROGRAMACIÓN**

**PROGRAMACIÓN III**

**APUNTE TEORICO**

**<<< >>>**

**APLICACIONES WINDOWS IV**

***Prof. ABEL OSCAR FAURE***

# APLICACIONES WINDOWS IV

## MENUS Y TOOLBARS

Hoy en día casi todas las aplicaciones para Windows poseen menús y barra de herramientas, Visual Studio a partir de 2005 proporciona estos controles.

Los dos controles que se van a ver representan un impulso atractivo del poder a los desarrolladores casuales y profesionales por igual. Crear aplicaciones con aspecto profesionales barras de herramientas y menús que antes eran utilizadas por programadores que se tomaban el tiempo para escribir controladores personalizados y los que han comprado los componentes de terceros. Lo que antes podía tardar semanas es ahora una tarea sencilla que, literalmente, se puede hacer en cuestión de segundos.

Los controles que se utilizan se pueden agrupar en una familia de controles que tiene la Franja de sufixo. Ellos son el ToolStrip, MenuStrip y StatusStrip. En su forma más pura, ToolStrip y MenuStrip son de hecho, el mismo control, porque se deriva MenuStrip directamente desde el ToolStrip. Esto significa que cualquier cosa que el ToolStrip puede hacer, el MenuStrip puede hacerlo. Obviamente, esto también significa que los dos trabajan muy bien juntos.

## EL CONTROL MENUSTRIP

En adición al control MenuStrip existen varios controles adicionales que se utilizan para rellenar un menú. Los tres más comunes son el ToolStripMenuItem, ToolStripDropDown y el ToolStripSeparator. Todos estos controles representan una manera particular de ver un elemento en un menú o una barra de herramientas. El ToolStripMenuItem representa una única entrada en un menú, el ToolStripDropDown representa un elemento que al hacer clic muestra una lista de otros artículos, y el ToolStripSeparator representa una línea horizontal de división o vertical de un menú o una barra de herramientas.

Hay otro tipo de menú que se discute brevemente el ContextMenuStrip. Un menú contextual aparece cuando un usuario hace clic en un elemento, y muestra por lo general la información pertinente a ese artículo.

Este primer ejemplo simplemente muestra lo fácil que es trabajar con estos controles.

1. Crear una nueva aplicación para Windows.
2. Arrastre una instancia del control MenuStrip del Cuadro de herramientas hasta la superficie de diseño.
3. Haga clic en el triángulo situado a la extrema derecha de la MenuStrip en la parte superior del cuadro de diálogo para mostrar las acciones ventana.
4. Haga clic en el pequeño triángulo en la esquina superior derecha del menú y haga clic en los elementos estándar de inserción enlace.

Eso es todo. Si desplegamos el menú Archivo, verá que se ha rellenado con todos los entradas, incluyendo atajos de teclado e iconos. No hay ninguna funcionalidad detrás del menú – de eso nos encargaremos más adelante.

## CREAR MENUS MANUALMENTE

Arrastrar el control MenuStrip del Cuadro de herramientas al Form y verás que este control se coloca a sí mismo tanto en el propio formulario y en la bandeja de control, pero se pueden editar directamente en el formulario. A crear nuevos elementos de menú, basta con colocar el puntero en la casilla Escriba aquí. Al entrar en el título del menú en el cuadro resaltado, puede incluir un signo (&) delante de una letra que desea actuar como el carácter clave de acceso directo para el elemento de menú - este es el carácter que aparece subrayado en el elemento de menú y que se puede seleccionar presionando Alt y la tecla (carácter) juntos. Es posible crear varios elementos del menú en el mismo menú con el carácter de tecla de acceso mismo. La regla es que un carácter se puede utilizar para este fin sólo una vez por cada menú desplegable (por ejemplo, una vez en los archivos en el menú emergente, una vez en el menú Ver, y así sucesivamente).

Si accidentalmente asigna el mismo carácter de tecla de acceso a opciones de menú múltiple en el mismo menú emergente, sólo el que está más cerca a la parte superior del control responde al carácter. Cuando se selecciona el tema, el control automáticamente muestra los elementos de relación con el tema actual y a la derecha del mismo. Cuando se introduce un nombre en cualquiera de estos controles, se crea un elemento nuevo en relación con el que empezó. Esa es la forma de crear los menús desplegables.

Para crear las líneas horizontales que dividen a los menús en grupos, se debe utilizar el ToolStripSeparator controlar en lugar de la ToolStripMenuItem, pero en realidad no se inserta un control diferente. En su lugar, sólo tiene que escribir un " - " (guión) como el único nombre de la leyenda del tema y Visual Studio a continuación supone automáticamente que el elemento es un separador y cambia el tipo de control.

En el siguiente ejercicio, crearemos un menú sin necesidad de utilizar Visual Studio para generar los elementos de la misma.

En este ejemplo, vamos a crear el menú archivo y los menús de ayuda desde el principio. Los menús Edición y Herramientas se dejan para que usted pueda hacerlo por sí mismo.

1. Crear un nuevo proyecto de aplicación Windows, el nombre menús manual.
2. Arrastre un control MenuStrip del Cuadro de herramientas hasta la superficie de diseño del Formulario.
3. Haga clic en el área de texto del control MenuStrip donde dice Escriba aquí, escriba & File y pulse el Tecla Enter.
4. Escriba lo siguiente en las áreas de texto debajo del elemento File:

- &New
- &Open
- -
- &Save
- Save &As
- -
- &Print
- Print Preview
- -
- E&xit

Observe cómo los guiones son cambiados automáticamente por Visual Studio a una línea que separa los elementos.

5. Haga clic en el área de texto a la derecha del menú Archivos y escriba &Help.

6. Escriba lo siguiente en las áreas de texto debajo del elemento Help:

- Contents
- Index
- Search
- -
- About

7. Volver al menú File y configurar las teclas de acceso directo para los elementos. Para ello, seleccione el elemento que desea configurar y encontrar la propiedad Teclas de método abreviado (ShortcutKeys) en las propiedades de panel. Luego hacer clic en la flecha del menú desplegable, se le presentará una pequeña ventana donde se puede establecer la combinación de teclas que desea asociar con el elemento de menú. Debido a que este es un menú estándar, se debe utilizar las combinaciones de teclas estándar, pero si va a crear otra cosa, no dude en seleccionar cualquier otra combinación de tecla. Establezca las propiedades Teclas de método abreviado en el menú File, como se muestra en la siguiente tabla:

ITEM NAME	PROPERTIES AND VALUES
&New	Ctrl+N
&Open	Ctrl+O
&Save	Ctrl+S
&Print	Ctrl+P

8. Ahora colocar alguna imagen al ítem del menú. Seleccione el elemento New en el menú File y haga clic en los puntos suspensivos (...) a la izquierda de la propiedad de imagen en el panel Propiedades para abrir la Selección De recursos. Podría decirse que lo más difícil de crear estos menús es la obtención de las imágenes que desea para mostrar.

9. Colocar la imagen para los demás ítems.

10. Ejecute el proyecto. Puede seleccionar el menú File, haga clic en ella o pulsando Alt + F, y usted puede acceder al menú de Ayuda con Alt + H.

## PROPIEDADES DEL ToolStripMenuItem

PROPIEDADES	DESCRIPCIÓN
Checked	<i>Indica si el menú esta seleccionado.</i>
CheckOnClick	Cuando esta propiedad es true, una marca o bien se agrega o se quita de la posición a la izquierda del texto en el elemento que está ocupado por otro modo una imagen. Utilice la propiedad Checked para determinar el estado del elemento del menú.
Enabled	<i>Si se establece en false, el elemento aparecerá en gris.</i>
DropDownItems	Devuelve una colección de elementos que se utiliza como un menú desplegable en relación con el elemento de menú.

## AGREGAR FUNCIONALIDAD AL MENU

La única tarea que queda por hacer es hacer que algo suceda cuando se hace clic en ellos. Crear un ejemplo muy sencillo que se basa en el anterior ejemplo. Para responder a las selecciones realizadas por el usuario, debe implementar los controladores de uno de los dos eventos que los ToolStripMenuItems envían:

EVENTOS	DESCRIPCIÓN
Click	<i>Se dispara cuando el usuario hace Click en un ítem.</i>
CheckedChanged	Se envía cuando un elemento con la propiedad CheckOnClick se hace clic.

## PRACTICA 10

1. Continuar utilizando el formulario que ha creado en el paso anterior. Arrastre un Rich-TextBox en el diseño de la superficie y cambiar su nombre por el de richTextBoxText. Establezca su propiedad Dock en Fill.
2. Seleccione el MenuStrip y luego Ingresar Format en el área de texto junto a la opción del menú Help y presione la tecla Enter.
3. Seleccione la opción de menú Format y arrastrarlo a una posición entre File y Help.
4. Agregar un ítem al menú Format con el texto Show Help menú.
5. Establezca la propiedad CheckOnClick del ítem Show Help menú a true. Establezca su propiedad Checked en true.

6. Seleccione showHelpMenuToolStripMenuItem y agregar un controlador de eventos para el evento CheckedChanged haciendo doble click en el evento de la lista de Eventos del panel Propiedades.

7. Agregar el siguiente código al evento:

```
private void showHelpMenuToolStripMenuItem_CheckedChanged(object sender, EventArgs e)
{
    ToolStripMenuItem item = (ToolStripMenuItem)sender;
    helpToolStripMenuItem.Visible = item.Checked;
}
```

8. Hacer doble click en newToolStripMenuItem, saveToolStripMenuItem y openToolStripMenuItem. Al hacer doble click en un ToolStripMenuItem en vista de diseño hace que el evento Click que se añada al código automáticamente. Agregar el siguiente código:

```
private void newToolStripMenuItem_Click(object sender, EventArgs e)
{
    richTextBoxText.Text = "";
}

private void openToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        richTextBoxText.LoadFile(@"Example.rtf");
    }
    catch { }
}

private void saveToolStripMenuItem_Click(object sender, EventArgs e)
{
    try
    {
        richTextBoxText.SaveFile("Example.rtf");
    }
    catch { }
}
```

9. Ejecutar la aplicación. Al hacer click en el ítem Shown Help menú, el menú Ayuda desaparece o aparece, dependiendo del estado de la propiedad Checked, y debe ser capaz de abrir, guardar y borrar el texto en el cuadro de texto.

## ¿Cómo funciona?

El evento showHelpMenuToolStripMenuItem\_CheckedChanged se manipula en primer lugar. El controlador de eventos para este caso establece la propiedad Visible de MenuItemHelp en true si la propiedad Checked es true, de lo contrario, debe ser false.

Esto hace que el ítem de menú se comporte como un botón de activación para el menú Help. Por último, los tres controladores de eventos para los eventos Click borrar el texto en RichTextBox, guardar el texto en el control RichTextBox en un archivo predeterminado, y abrir dicho fichero, respectivamente.

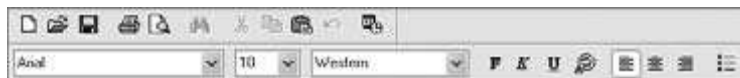
Tenga en cuenta que el click y Eventos CheckedChanged son idénticos en tanto que controlará el evento que se produce cuando un usuario hace click en un ítem de me-

nú, pero el comportamiento de los ítems del menú en cuestión son muy diferentes y deben ser manejados de acuerdo con el propósito de la opción del menú.

## EL CONTROL TOOLBARS

Mientras que los menús son buenos para facilitar el acceso a una gran cantidad de funcionalidad en las aplicaciones, también nos podemos beneficiar colocando algunas funcionalidades en una barra de herramientas. Una barra de herramientas proporciona acceso con un click a estas funciones de uso frecuente como Abrir, Guardar, y así sucesivamente.

La siguiente figura muestra una selección de las barras de herramientas en WordPad.



Un botón en la barra de herramientas por lo general muestra una imagen y texto, aunque es posible tener botones con ambos. Si deja el puntero del ratón flotar por encima de un botón en una barra de herramientas, a menudo se mostrará una descripción, que proporciona información sobre el propósito del botón, sobre todo cuando sólo se muestra un icono.

El ToolStrip, al igual que el MenuStrip, se ha hecho con un aspecto profesional. Cuando los usuarios ven una barra de herramientas, esperan ser capaces de moverse y colocarla donde se quiera. La ToolStrip permite a los usuarios precisamente hacer eso. La primera vez que agrega un ToolStrip a la superficie de diseño del formulario se ve muy similar a la MenuStrip que se mostró anteriormente, excepto por dos cosas: En el extremo izquierdo son cuatro puntos verticales, tal y como los conocemos por los menús de Visual Studio. Estos puntos indican que la barra de herramientas se pueden mover y acoplar en la ventana de aplicación principal. La segunda diferencia es que por defecto una barra de herramientas muestra las imágenes, en lugar de texto, por lo que el valor por defecto de los elementos de la barra es un botón. La barra de herramientas muestra una lista desplegable menú que permite seleccionar el tipo de elemento.

### Propiedades ToolStrip

PROPIEDADES	DESCRIPCIÓN
GripStyle	Controla si los cuatro puntos verticales se muestran en el extremo izquierdo de la barra de herramientas. El efecto de ocultar el agarre es que los usuarios ya no pueden mover la barra de herramientas.
LayoutStyle	Controla cómo los elementos de la barra de herramientas se muestran. El valor predeterminado es horizontal.
ítems	<i>Devuelve una colección de todos los ítems del ToolBars</i>
ShowItemToolTip	Determina si se debe mostrar información sobre herramientas para los elementos de la barra de herramientas.

Stretch

De forma predeterminada, una barra de herramientas es sólo ligeramente más ancho o más alto que los elementos contenidos dentro de ella. Si establece la propiedad Stretch en true, la barra de herramientas ocupara toda la longitud de su contenedor.

### ***ToolStrip ítems***

Anteriormente, se mencionó que una barra de herramientas debe ser capaz de contener botones, cuadros combinados y cuadros de texto. Como era de esperar, hay controles para cada uno de estos ítems, pero también hay otros, que se describen en la siguiente tabla:

CONTROL	DESCRIPCIÓN
ToolStripButton	Representa un botón. Usted puede usar este para los botones, con o sin texto.
ToolStripLabel	Representa una etiqueta. También puede mostrar imágenes, lo que significa que este control se puede utilizar para mostrar una imagen estática delante de otro control que no se muestra información sobre sí mismo, como un cuadro de texto o cuadro combinado.
ToolStripSplitButton	Muestra un botón con un botón de lista desplegable a la derecha, que cuando se hace clic, se muestra un menú por debajo de ella. El menú no se despliega si el botón es parte del control que hace click.
ToolStripDropDownButton	Similar a la ToolStripSplitButton. La única diferencia es que el botón desplegable ha sido eliminado y reemplazado con una imagen de una flecha hacia abajo.
ToolStripComboBox	<i>Muestra un combo box.</i>
ToolStripProgressBar	Incorpora una barra de progreso en la barra de herramientas.
ToolStripTextBox	<i>Muestra un Text Box.</i>
ToolStripSeparator	Crea divisiones horizontales o verticales de los elementos.



## PRACTICA 11

- Continuar trabajando con el ejemplo anterior y eliminar el ToolStripMenuItem que se usó en el menú Format. Seleccione la opción Show Help menú y pulse la tecla Suprimir. Añadir tres ToolStripMenuItems en su lugar y cambiar cada uno de sus CheckOnClick propiedades a true:
  - Bold.
  - Italic.
  - Underline.
- Añadir un ToolStrip al formulario. En la ventana de Acciones, haga clic en Insertar elementos estándar. Seleccionar y eliminar los ítem para Cortar, Copiar, Pegar, y el separador de ellos. Cuando se inserta el ToolStrip, el RichTextBox puede dejar de mostrarse correctamente. Si eso ocurre, cambiar el estilo Dock a ninguno y de forma manual cambiar el tamaño del control para rellenar el formulario. A continuación, cambie la propiedad Anchor en Superior, Inferior, Izquierda, Derecho.
- Crear tres nuevos botones y un separador al final de la barra de. (Haga clic en el último elemento en la ToolStrip para abrir las opciones.)
- Crear los dos últimos seleccionando ComboBox en la lista desplegable y luego añadir uno de separación como el último elemento.
- Seleccione el elemento de ayuda y se arrastra desde su posición actual a la posición como el último elemento de la barra de herramientas.
- Los primeros tres botones van a ser los botones de negrita, cursiva y subrayado, respectivamente. El Nombre de cada uno se muestra en la siguiente tabla:

TOOLSTRIPBUTTON	NOMBRE
Bold button	boldToolStripButton
Italic button	italicToolStripButton
Underline button	underlineToolStripButton
ComboBox	fontsToolStripComboBox

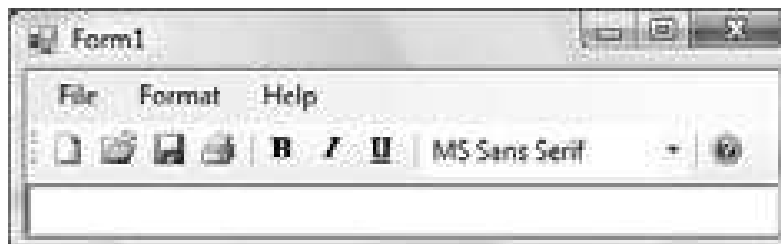
- Seleccione el botón Negrita, haga clic en los puntos suspensivos (...) en la propiedad de imagen y cargue la imagen que corresponda.
- Realizar el paso 8 para Italic y Underline.
- Seleccione el ToolStripComboBox. En el panel Propiedades, establezca las propiedades que se muestran en el siguiente tabla:

PROPIEDADES	VALOR
ítems	MS Sans Serif Times New Roman
DropDownStyle	DropDownList

10. Establezca la propiedad CheckOnClick para cada uno de los botones Bold, Italic y Subrayado en true.
11. Para seleccionar el elemento inicial en el ComboBox, escriba lo siguiente en el constructor de la clase:

```
public Form1()
{
    InitializeComponent();
    fontsToolStripComboBox.SelectedIndex = 0;
}
```

12. Correr aplicación.



13. Ya tenemos los controladores de eventos para Save, New y Open cuando se llaman del menú y obviamente los botones en la barra de herramientas que se comportan exactamente de la misma manera que los elementos del menú. Esto se consigue fácilmente, mediante la asignación de los eventos Click de los botones de las barras de herramientas a los mismos controladores que son utilizados por los elementos del menú. Establecer los mismos de la siguiente manera:

TOOLSTRIPBUTTON	EVENTO
New	newToolStripMenuItem_Click
Open	openToolStripMenuItem_Click
Save	saveToolStripMenuItem_Click

14. Ahora es el momento de agregar los controladores de eventos para los botones Bold, Italic y Underline. Estos botones pueden ya estar seleccionados (Checked) por eso utilizamos el evento CheckChanged en lugar del evento Click. Añadir el evento CheckChange para cada uno de los tres botones y Agregue el siguiente código:

```
private void boldToolStripButton_CheckedChanged(object sender, EventArgs e)
{
    Font oldFont, newFont;
    bool checkState = ((ToolStripButton)sender).Checked;
    oldFont = this.richTextBoxText.SelectionFont;
    if (!checkState)
        newFont = new Font(oldFont, oldFont.Style & ~FontStyle.Bold);
    else
        newFont = new Font(oldFont, oldFont.Style | FontStyle.Bold);
    richTextBoxText.SelectionFont = newFont;
    richTextBoxText.Focus();
    boldToolStripMenuItem.CheckedChanged -= new
    EventHandler(boldToolStripMenuItem_CheckedChanged);
    boldToolStripMenuItem.Checked = checkState;
    boldToolStripMenuItem.CheckedChanged += new
    EventHandler(boldToolStripMenuItem_CheckedChanged);
}

private void italicToolStripButton_CheckedChanged(object sender, EventArgs e)
{
    Font oldFont, newFont;
    bool checkState = ((ToolStripButton)sender).Checked;
    oldFont = this.richTextBoxText.SelectionFont;
    if (!checkState)
        newFont = new Font(oldFont, oldFont.Style & ~FontStyle.Italic);
    else
        newFont = new Font(oldFont, oldFont.Style | FontStyle.Italic);
    richTextBoxText.SelectionFont = newFont;
    richTextBoxText.Focus();
    italicToolStripMenuItem.CheckedChanged -= new
    EventHandler(italicToolStripMenuItem_CheckedChanged);
    italicToolStripMenuItem.Checked = checkState;
    italicToolStripMenuItem.CheckedChanged += new
    EventHandler(italicToolStripMenuItem_CheckedChanged);
}

private void underlineToolStripButton_CheckedChanged(object sender, EventArgs e)
{
    Font oldFont, newFont;
    bool checkState = ((ToolStripButton)sender).Checked;
    oldFont = this.richTextBoxText.SelectionFont;
    if (!checkState)
        newFont = new Font(oldFont, oldFont.Style & ~FontStyle.Underline);
    else
        newFont = new Font(oldFont, oldFont.Style | FontStyle.Underline);
    richTextBoxText.SelectionFont = newFont;
    richTextBoxText.Focus();
    underlineToolStripMenuItem.CheckedChanged -= new
    EventHandler(underlineToolStripMenuItem_CheckedChanged);
    underlineToolStripMenuItem.Checked = checkState;
    underlineToolStripMenuItem.CheckedChanged += new
    EventHandler(underlineToolStripMenuItem_CheckedChanged);
}
```

Los controladores de eventos anteriores, simplemente definieron el estilo correcto para el tipo de letra utilizado en el control RichTextBox.

Las tres últimas líneas en cada uno de los tres métodos se relaciona con el elemento correspondiente en el menú. La primera línea elimina el control de eventos en el elemento de menú. Esto asegura que ningún evento se dispara cuando se ejecuta la siguiente lí-

nea, que establece el estado de la propiedad Checked para el mismo valor que el botón de la barra. Por último, se restablece el controlador de eventos.

- 15.** Los controladores de eventos para los elementos del menú, simplemente deben establecer la propiedad Checked de los botones de la barra de herramientas, permitiendo a los controladores de eventos de los botones de la barra de herramientas hacer el resto. Agregar los controladores de eventos para el evento CheckChanged y escriba este código:

```
private void boldToolStripMenuItem_CheckedChanged(object sender, EventArgs e)
{
    boldToolStripButton.Checked = boldToolStripMenuItem.Checked;
}

private void italicToolStripMenuItem_CheckedChanged(object sender, EventArgs e)
{
    italicToolStripButton.Checked = italicToolStripMenuItem.Checked;
}

private void underlineToolStripMenuItem_CheckedChanged(object sender, EventArgs e)
{
    underlineToolStripButton.Checked = underlineToolStripMenuItem.Checked;
}
```

- 16.** Lo único que queda por hacer es permitir a los usuarios seleccionar una familia de fuentes del ComboBox. Siempre que un usuario cambia la selección en el cuadro combinado, el evento SelectedIndexChanged se dispara, por lo que hay que añadir un controlador de eventos para SelectedIndexChanged:

```
private void fontsToolStripComboBox_SelectedIndexChanged(object sender, EventArgs e)
{
    string text = ((ToolStripComboBox)sender).SelectedItem.ToString();
    Font newFont = null;
    if (richTextBoxText.SelectionFont == null)
        newFont = new Font(text, richTextBoxText.Font.Size);
    else
        newFont = new Font(text, richTextBoxText.SelectionFont.Size,
            richTextBoxText.SelectionFont.Style);
    richTextBoxText.SelectionFont = newFont;
}
```

- 17.** Ahora ejecuta el código. Usted será capaz de establecer en negrita y cursiva, y subrayado el texto desde la barra de herramientas. Tenga en cuenta que cuando se active o desactive un botón en la barra de herramientas, el elemento correspondiente en el menú estará marcado o sin marcar.

## EL CONTROL STATUSSTRIP

El último de la pequeña familia de controles de la banda es el StatusStrip. Este control representa la barra que se encuentra en la parte inferior del cuadro de diálogo en muchas aplicaciones. La barra se utiliza normalmente para mostrar brevemente información sobre el estado actual de la aplicación - un buen ejemplo es la pantalla de Word de la página actual, columnas, líneas, y así sucesivamente. El StatusStrip se deriva de ToolStrip.

PROPIEDADES	DESCRIPCIÓN
AutoSize	AutoSize está activada por defecto, que en realidad no es muy intuitivo, ya que no lo hacen desea que las etiquetas en la barra de estado para saltar adelante y atrás sólo porque se cambió el texto de uno de ellos. A menos que la información de la etiqueta es estática, siempre cambiar esta propiedad a false.
DoubleClickEnable	Especifica si el evento DoubleClick se va a disparar.

1. Cree una barra de estado simple para el ejemplo en el que hemos estado trabajando. la barra de estado cuenta con cuatro paneles, tres de los cuales muestran una imagen y texto, el último panel muestra sólo el texto.
2. Haga doble clic en el StatusStrip en el cuadro de herramientas para agregarlo al cuadro de diálogo. Puede que tenga que cambiar el tamaño del RichTextBox en el formulario.
3. En el panel Propiedades, haga clic en los puntos suspensivos (...) en la propiedad ítems del StatusStrip. Aparecerá el Editor de la colección de elementos.
4. Haga clic en el botón Agregar para agregar cuatro paneles a la StatusStrip. Establezca las siguientes propiedades en los paneles:

PANEL	PROPIEDAD	VALOR
1	Name	toolStripStatusLabelText
	Text	Clear this property
	AutoSize	False
	DisplayStyle	Text
	Font	Arial; 8.25pt; style=Bold
	Size	259, 17
	TextAlign	Middle Left

2	Name	toolStripStatusLabelBold
	Text	Bold
	DisplayStyle	ImageAndText
	Enabled	False
	Font	Arial; 8.25pt; style=Bold
	Size	47, 17
	Image	BLD
	ImageAlign	Middle-Center
3	Name	toolStripStatusLabelItalic
	Text	Italic
	DisplayStyle	ImageAndText
	Enabled	False
	Font	Arial; 8.25pt; style=Bold
	Size	48, 17
	Image	ITL
	ImageAlign	Middle-Center
4	Name	toolStripStatusLabelUnderline
	Text	Underline
	DisplayStyle	ImageAndText
	Enabled	False
	Font	Arial; 8.25pt; style=Bold
	Size	76, 17
	Image	UNDRLN
	ImageAlign	Middle-Center

5. Agregar la siguiente línea de código al final del evento `boldToolStripButton_CheckedChanged`:

```
toolStripStatusLabelBold.Enabled = checkState;
```

6. Agregar la siguiente línea de código al final del evento italicToolStripButton\_CheckedChanged:

```
toolStripStatusLabelItalic.Enabled = checkState;
```

7. Agregar la siguiente línea de código al final del evento underlineToolStripButton\_CheckedChanged:

```
toolStripStatusLabelUnderline.Enabled = checkState;
```

8. Seleccionar el RichTextBox y agregar el evento TextChanged luego colocar el siguiente código:

```
private void richTextBoxText_TextChanged(object sender, EventArgs e)
{
    toolStripStatusLabelText.Text = "Number of characters: " +
    richTextBoxText.Text.Length;
}
```

9. Correr aplicación.

