





Índice

1.	Menús usuarios.....	2
1.1.	MainMenu – Menú principal – Menu 	2
1.2.	ContextMenu – Menú Contextual – ctx 	4
1.3.	ToolStrip – Barra de herramientas – tb 	5
1.4.	StatusStrip – Barra de Estado – sts 	7

1. Menús usuarios

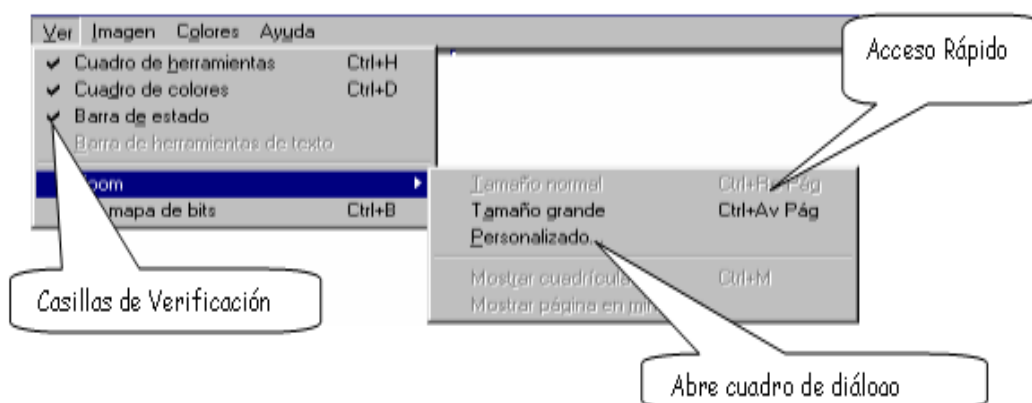
Principalmente lo que se pretende es mejorar el interface del usuario, para ello le crearemos menús desplegables, contextuales, barras de herramientas y barras de estado.

1.1. MainMenu – Menú principal – Menu



Para crear un Menú principal, debes ir introduciendo todos los elementos que quieras que aparezcan en el menú. Incluido los separadores.

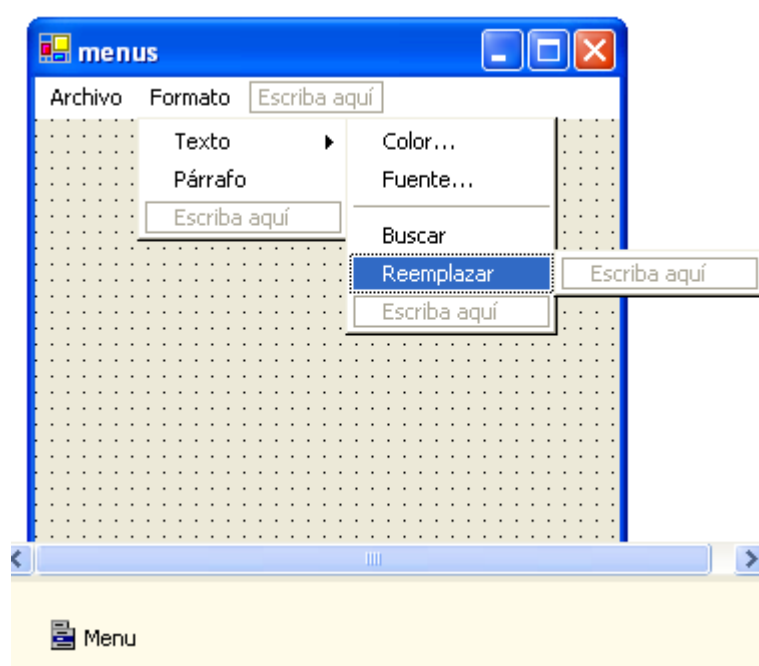
Sin olvidar los estándares:



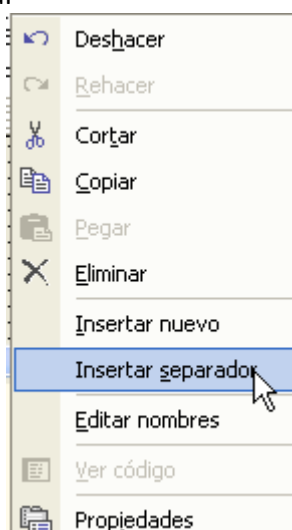
Como muestra la imagen:

- Algunas operaciones pueden realizarse a través del teclado, pulsando las teclas correspondientes a un Atajo o Acceso Rápido. O también pulsando Alt + Tecla subrayada (Ej. Alt+V → Desplegaría el menú Ver)
- Aquellos elementos del menú que terminan en puntos suspensivos ... me está indicando que al pulsar en ellos se abrirá una ventana de diálogo
- Algunos a su izquierda puede aparecer una tick, viset, o casilla de verificación, para indicar si están activados o no. Si en vez de un tick aparece un punto, me indica que de los elementos de ese bloque, sólo podría elegir uno.
- Cuando a la derecha me encuentro una flechita, me indica que al pinchar en ese elemento se me desplegará un nuevo menú.

Entonces para crear un menú como el de la imagen:

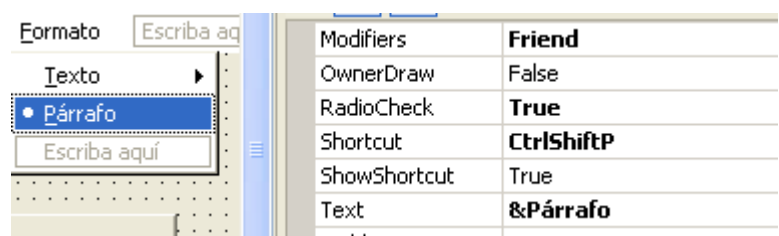


- Debes indicar en el formulario, en su propiedad MainMenuStrip, indicar el nombre (Name) del menú.
- Ir situándote en Escriba Aquí, e ir introduciendo los distintos Textos que quieres que el usuario lea cuando despliega los menús.
- De igual forma podrás introducir separadores, o eliminar elementos del menú, desplegando el menú contextual

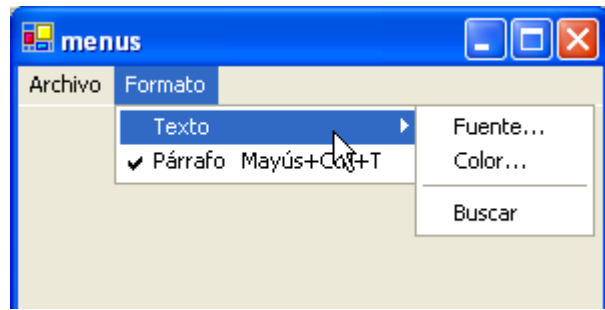


- Por cada elemento del menú, debemos configurar sus propiedades, destacando:
 - **Name:** Nombre que emplearemos para referirnos a él en el código. Generalmente: mnuTextoVisible
 - **Enabled:** Indica si está o no activo para el usuario.

- **Text:** Texto que el usuario ve. Si queremos que exista una letra de acceso directo pulsando Alt+ letra subrayada, en esta propiedad escribes & delante de la letra que quieras señalar.
- **Shortcut:** Para indicar la combinación de teclas que me permitirá ejecutar directamente este comando.
- **ShowShortcut:** a true, para que me muestre la combinación de teclas elegida
- **RadioCheck:** Si queremos que aparezca señalado con un punto cuando la propiedad **checked** esté a true. En caso de que RadioCheck esté a false, cuando checked esté a true, en vez de un punto, aparecerá un tick o viset.



Al ejecutarse con RadioCheck a false:



En este punto, una vez que le hemos rellenado Name de todos los comandos de todos los menús, y hemos configurado las propiedades a nuestro gusto, solo quedaría introducir el código que se ejecutará cuando el usuario haga clic en cualquiera de los elementos.

1.2. ContextMenu – Menú Contextual – ctx

Como todos sabemos, un menú contextual es aquel que se abre al hacer clic con el botón derecho sobre el control al que está asociado dicho menú contextual.

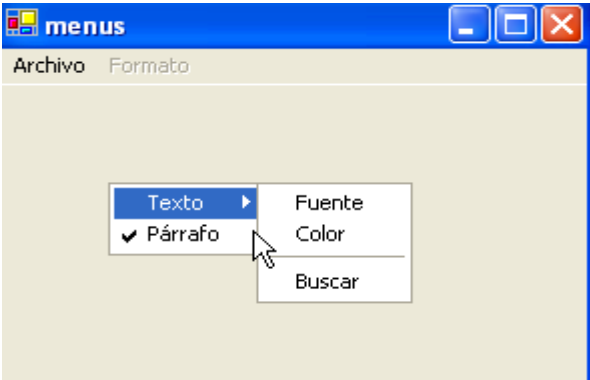
Para crearlo, seleccionamos en control ContextMenuStrip, a igual que el anterior, es un control no visible en el formulario, por lo que se queda fuera. Una vez que ya le hemos dado su nombre (Name), pasamos a indicar los elementos que queremos tener en él:



Como ves es de igual forma que el menú principal, a excepción de que el elemento principal, se llama Menú Contextual, y eso no se puede cambiar. A partir de él, podrás crearlo exactamente igual que el anterior.

Una vez creado todos los elementos del menú, con sus propiedades bien configuradas y el código de cada uno de los elementos, sólo nos quedaría vincularlo al elemento del que nos interesa sea menú contextual. Para ello, en la propiedad **ContextMenu** de dicho elemento, le indicamos el Name de nuestro menú.

Así cuando pulsemos con el botón derecho sobre ese control (en el ejemplo de la imagen es el Form), aparecerá el menú.



1.3. ToolStrip – Barra de herramientas – tb



Muy vinculado con los menús también se encuentra este otro control, las barras de herramientas, y en plural, porque puedo tener tantas como quiera.

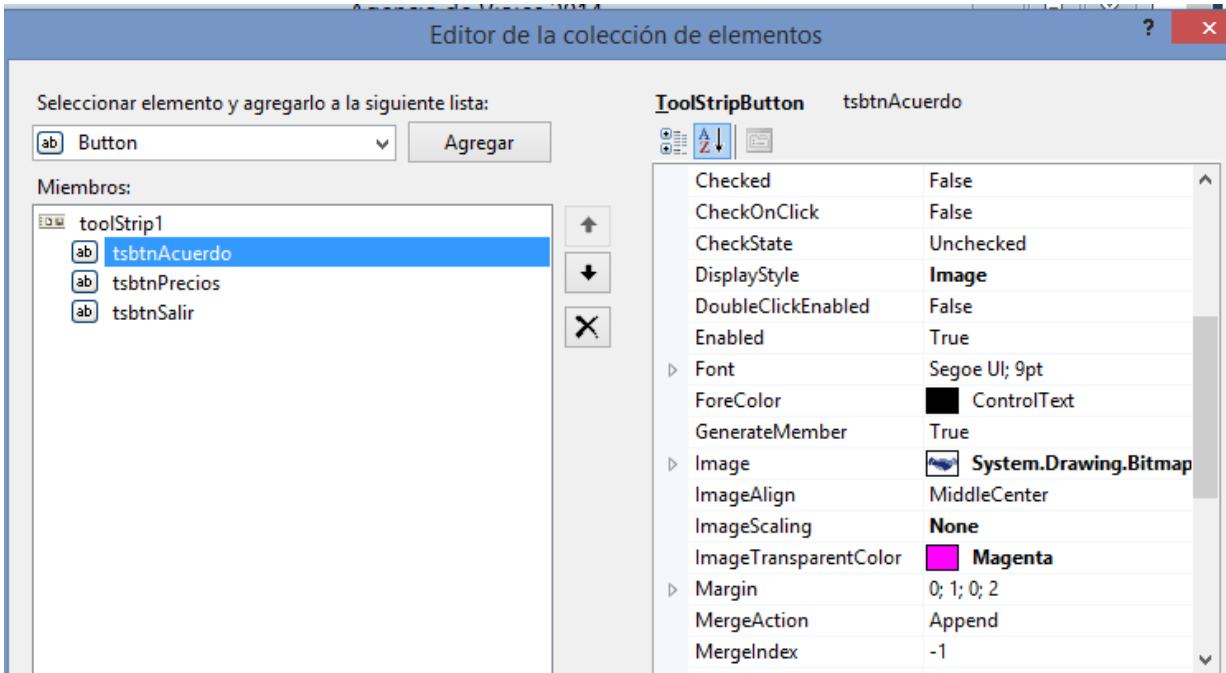
Sabemos que las barras de herramientas son aquellas donde se colocan los botones que ejecutan los comandos de los menús más utilizados. Dichos comandos, vienen representados generalmente, solo mediante un icono, aunque también lo asistamos con un **tooltip**, por si la imagen elegida no fuese del todo explicativa, aunque no quita, que podamos incluir texto en el botón.

Para crear una barra de herramientas elegimos el control y lo dibujamos en el formulario. Por defecto, su propiedad **Dock** estará a Top, por lo que se acoplará en la parte superior del formulario, pero podríamos cambiarlo.

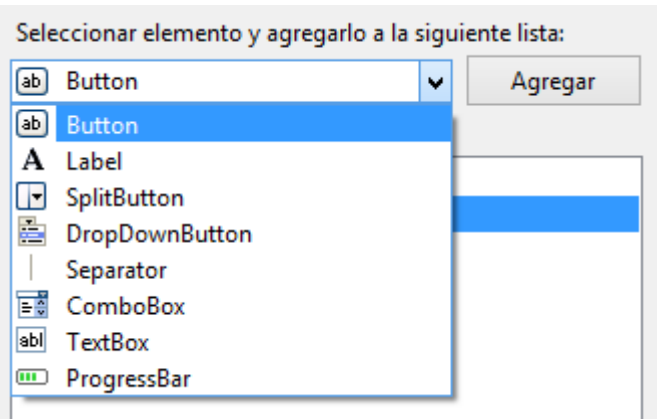
Antes de empezar a configurar los botones de la barra de herramientas, debemos recordar que lo suyo, es que cada botón lleve una imagen representativa. Por lo que es importante que busques imágenes representativas de lo que va a realizar ese botón.

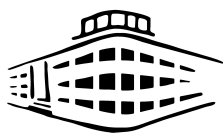

La propiedad principal de este control es **Items**, ya que es donde se definen los botones que va a contener. Como vemos es una colección por lo que tendremos que añadir uno a uno los botones y todas sus propiedades, entre ellas destacamos:

- **(Name):** Como no.
- **Enabled:** Indica si está o no en activo
- **Image:** Seleccionas el fichero imagen.
- **ImageScaling:** O **None** y se ajusta al tamaño de la imagen, o **SizeToFit** y la imagen se ajusta al tamaño del botón.
- **ToolTipText:** indicas el mensaje de ayuda complementario a la imagen seleccionada

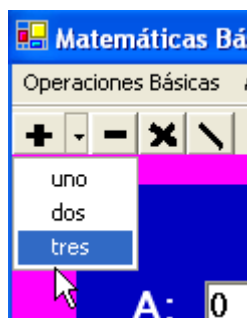


Podemos seleccionar los siguientes tipos de controles:



 CIPF VIRGEN DE GRACIA	TEMA 5 Aplicaciones Windows – Menús y Barras de Herramientas	 Página 7 de 8
--	---	---

Si elegimos un SplitButton, tendríamos algo similar al botón suma, actuando tanto como botón, como menú desplegable.



Para codificar los botones... Lo único es que podemos acceder a ellos directamente a través de su (Name) y ya gestionarlos como si estuviesen en medio de un formulario en vez de en una barra de herramientas.

Ejemplo de un botón SALIR de una ToolStrip

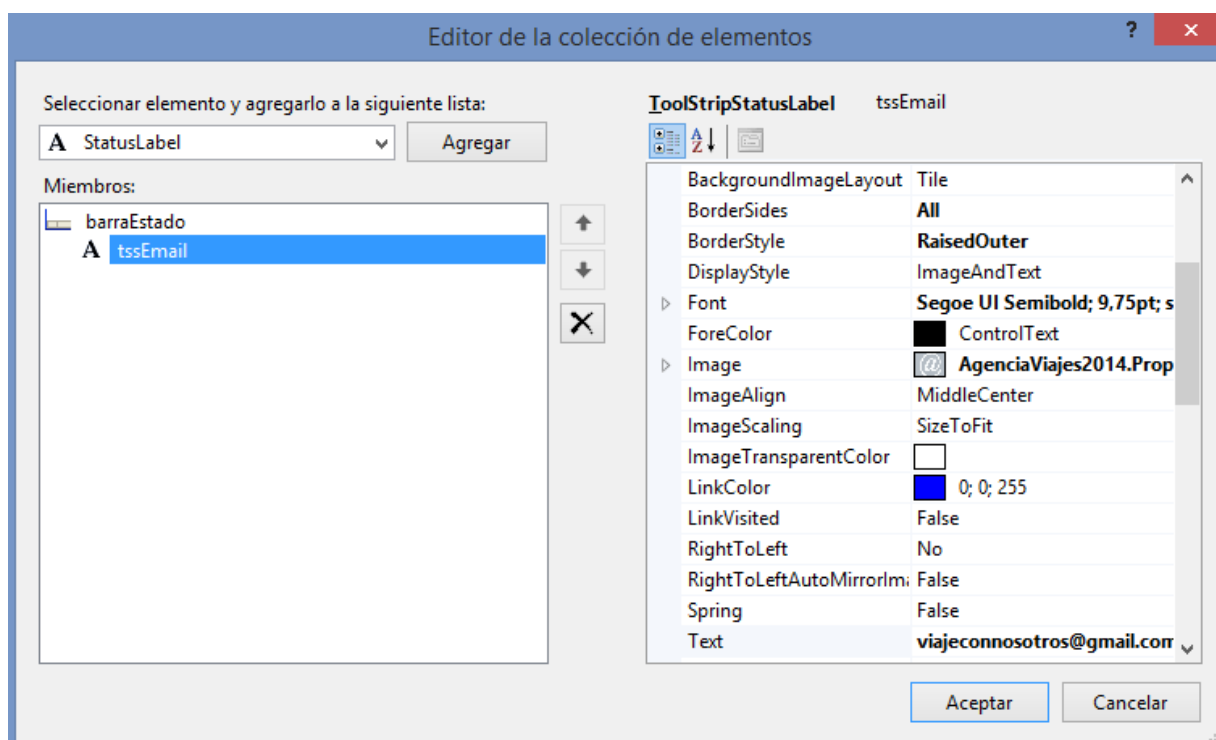
```
private void tsbtnSalir_Click(object sender, EventArgs e)
{
    this.Close();
}
```

1.4. StatusStrip – Barra de Estado – sts



Las barras de estado son aquellas que aparecen en la parte inferior de nuestra ventana, su propiedad **Dock** está a Bottom, generalmente para darnos información de las operaciones que está realizando mi aplicación, o bien, información del sistema, etc.

La información que queremos mostrar se crearán como elementos de mi Barra de Estado, y por tanto, habrá que añadirlo a su colección de elementos (**items**), teniendo cada uno sus propiedades particulares y autonomía.



Para añadir **items**, agregar cada uno de los items e ir configurando sus propiedades, entre las que destacaremos:

- Si es una etiqueta: **ToolStripStatusLabel**
 - **(Name):** Podremos referirnos en el código a él, directamente o como índice de la colección de items de la barra de herramientas (mucho más rollo)
 - **Spring:** A true, indicas que quieres que este elemento rellene el espacio sobrante, no todos los ítems deben tener esta propiedad a true.
 - **BorderSides:** Los lados del ítem que muestra los bordes (de ninguno a todos)
 - **BorderStyle:** Elijo el tipo de borde: Remarcado, hundido, simple, tc.
 - **DisplayStyle:** Si queremos mostrar imagen y/o texto
 - **Image:** Si queremos que se muestre un icono
 - **ImageScaling:** lo suyo es que se ajuste a la barra y tome el valor de **SizeToFit**
 - **IsLink:** Si queremos convertir una label en una **LinkLabel**.
 - **Text:** El texto que quiero que aparezca
 - **TextAlign:** Alineación del contenido
 - **TextImageRelation:** Especificamos cómo aparece la imagen con respecto al texto
 - **ToolTipText:** El texto de información al usuario, para que se muestre no solo debes escribir el texto, sino que en la propiedad **ShowItemsToolTip** del StatusStrip debe estar a true