

Desarrollo de Aplicaciones con Lenguajes Visuales

Docente / Martin Murdaca

Clase

6

UNIDAD 2

3er trimestre 2020

1. Presentación

Hola a todos. Si bien la mayoría adeuda trabajos, seguiremos adelante. Reitero la importancia de respetar las fechas de entrega. Les pido que me notifiquen ante cualquier situación problemática que les impidió cumplir en fecha con la entrega de los trabajos.

Les recuerdo nuevamente que es importante y **obligatorio ingresar al aula todas las semanas por lo menos una vez**, no sólo para leer y descargar todo el material de la semana sino también para interactuar o consultar dudas con los compañeros. También sigamos conectados a través de mi dirección de correo personal martinmurdaca@gmail.com.

En la clase de hoy estudiaremos cómo crear un menú desplegable. Para ello trabajaremos con el objeto **MenuStrip**. Este control nos permite crear un menú principal organizando comandos para que el usuario los ejecute, de la misma manera que utilizamos siempre la barra de menú de cualquier ventana de Windows.

También presentaremos los **cuadros de diálogo**, que invocan a las ventanas ya conocidas y manipuladas desde los programas que operan bajo Windows, tales como: cuadro de diálogo de Fuente, Guardar como, Colores, entre otros.



Además de los temas vertidos en la clase de esta semana, les recomiendo repasar el tema **Vectores**, ya que es un tema que ya vieron teóricamente en otros talleres y me gustaría que pudieran ponerlo en práctica con este lenguaje, tal como lo hicimos con las variables.

Desarrollo de los contenidos

1) Menús

El control que nos permite desde Visual Basic crear un menú principal organizando comandos para que el usuario los ejecute es el objeto **MenuStrip**.

Para crear un menú personalizado seleccionen este objeto desde la Ficha Menús y barras de herramientas.

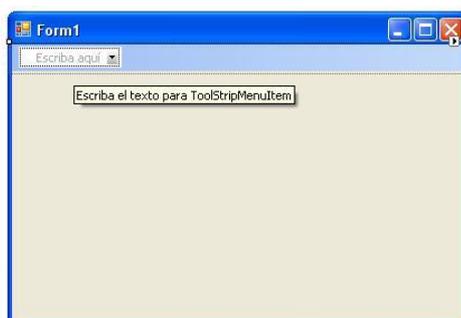


Figura 1
El objeto MenuStrip

La creación de las distintas opciones del menú es un proceso que se facilitó con relación a versiones anteriores. Simplemente se van editando cada opción del menú desde el mismo formulario.

Al seleccionar la primera opción podemos modificar sus propiedades. Luego, desplazándonos con el click del Mouse o con las flechas de movimiento del teclado iremos completando todo el menú desplegable.

A continuación mostraremos las propiedades más importantes que podemos modificar desde cada ítem del menú:

Tabla I Propiedades del Editor de Menú

Text	Nombre del comando que aparecerá en la barra de menú o en un menú desplegable asociado. Para permitir el acceso por teclado a dicho elemento, deberán anteponer un ampersand (&) a la letra que desea que aparezca subrayada, permitiendo al usuario acceder combinado la tecla ALT +la letra resaltada. Ej: &Archivo, se vería <u>A</u> rchivo.
Name	Nombre del control para el elemento del menú. Este, como vimos hasta ahora, es un nombre que identifica al objeto cuando este interviene en la programación. No para ser visualizado. Sin embargo, es obligatorio que asigne a cada elemento del menú un Nombre, ya que este no trae uno en forma predeterminada. Sería aconsejable que utilizara el prefijo mnu para nombrarlos.
ShorCutKeys	Presenta una lista, en la que podemos elegir cual será el método abreviado para acceder a dicho elemento. Esta aparecerá a la derecha del título del menú.

Text	Nombre del comando que aparecerá en la barra de menú o en un menú desplegable asociado. Para permitir el acceso por teclado a dicho elemento, deberán anteponer un ampersand (&) a la letra que desea que aparezca subrayada, permitiendo al usuario acceder combinado la tecla ALT +la letra resaltada. Ej: &Archivo, se vería <u>A</u> rchivo.
Checked	Es una casilla de verificación que ustedes la seleccionan, si desean que aparezca junto al menú una marca de verificación ubicada sobre su lado izquierdo.
Enabled	Permite seleccionar si el elemento del menú debe responder a eventos. Desactiven esta opción si desea que el elemento del menú no esté disponible y aparezca atenuado.
Visible	Permite hacer que un elemento aparezca o no en el menú. Pueden utilizarlo cuando desea que una opción de menú aparezca solo si el usuario realiza una acción determinada.

A. Elementos del Menú

Si bien el elemento predeterminado de un menú es el **MenuItem** existen otros tres tipos de elementos más que podemos seleccionar como parte de nuestro menú. Ellos son:

- **ComboBox**: inserta una lista desplegable como parte del menú.
- **Separator**: establece una línea que establece una separación entre las opciones del menú.
- **TextBox**: inserta un cuadro de texto que podrá ser editado en tiempo de ejecución.

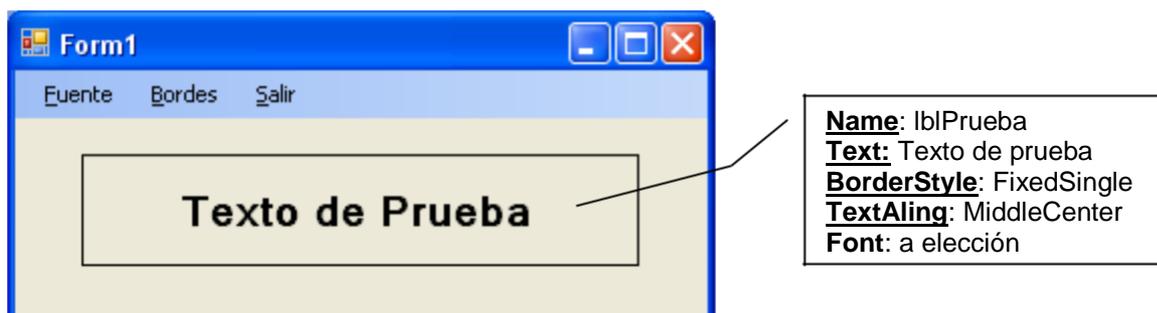
B. ¿Cómo se crea un Menú?

Ustedes crearán una aplicación nueva, la cual contendrá un menú de opciones que se detallará a continuación, y un objeto *label*, el cual sufrirá los cambios basado en las opciones elegidas por el usuario.

La siguiente figura, muestra la apariencia que tendrá el formulario:

Figura 2

Diseño del Formulario con la Barra de Menú



Y la tabla que sigue presenta el diseño que tendrá su nuevo menú.

Tabla II Ajustes del menú en tiempo de diseño

<i>Objeto del Menú</i>	<i>PropiedadValor en tiempo de diseño</i>	
Fuente	Name mnuFuente	
Color	NamemnuColor	
Automático	NamemnuAuto	Image: 
Rojo	NamemnuRojo	Image: 
Azul	Name mnuAzul	Image: 
Verde	NamemnuVerde	Image: 
Bordes	NamemnuBordes	
Bordes		
Ninguno	NamemnuNinguno	
Sencillo	NamemnuSencillo	
Efecto 3D	NamemnuEfecto3D	



Las imágenes podrán encontrarlas en C:\Archivos de programa\Microsoft Office\MEDIA\OFFICE11\BULLETS.

Observen que nombramos los elementos del menú con un prefijo “mnu” seguido a la leyenda que identifica la acción que podremos llevar a cabo con dicho menú. Si bien esto no es obligatorio, es de gran utilidad para el momento en que codifique dicho menú.

La idea del programa consiste en aplicar diferentes formatos de color de fuente y bordes, al objeto llamado lblTexto, que nos servirá como texto de muestra.

Comenzaremos por el menú Fuente [→] Color. Lo único que deben conocer es el valor de la propiedad que representa el color de la letra.

Como ya lo utilizamos en capítulos anteriores, Visual Basic tiene una estructura de colores que podemos utilizar para ello.

```
Private Sub mnuAuto_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuAuto.Click
    lblPrueba.ForeColor = Color.Black
```

End Sub

Private Sub mnuRojo_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuRojo.Click

 lblPrueba.ForeColor = **Color.Red**

End Sub

Private Sub mnuAzul_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuAzul.Click

 lblPrueba.ForeColor = **Color.Blue**

End Sub

Private Sub mnuVerde_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuVerde.Click

 lblPrueba.ForeColor = **Color.Green**

End Sub

Private Sub mnuNinguno_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles mnuNinguno.Click

 lblPrueba.BorderStyle = **BorderStyle.None**

End Sub

Private Sub mnuSencillo_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuSencillo.Click

 lblPrueba.BorderStyle =

BorderStyle.FixedSingle End Sub

Private Sub mnuEfecto3D_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles mnuEfecto3D.Click

 lblPrueba.BorderStyle = **BorderStyle.Fixed3D**

End Sub

Private Sub mnuSalir_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles mnuSalir.Click

End

End Sub**C. Menú Contextual**

Mediante el control **ContextMenuStrip** se puede crear un menú contextual, que aparecerá al pulsar el botón derecho del Mouse sobre el objeto al que fue asociado.

En primer lugar agregaremos este control al formulario, y podrán observar que este menú quedará oculto para el usuario difiriendo del control visto anteriormente, el objeto **MenuStripControl**. Lo llamaremos **mnuContextual**.

Agregaremos las siguientes opciones:

<i>Objeto del Menú</i>	<i>PropiedadValor en tiempo de diseño</i>
------------------------	---

Sin Borde	NamemnuSinBorde
Borde Sencillo	NamemnuBordeSen

Como ambos elementos ya los programamos anteriormente, solo diseccionaremos al evento click de esos elementos del menú anterior. Así quedarán codificados:

```
Private Sub SinBorde_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuSinBorde.Click
```

```
    mnuNinguno_Click(sender, e)
```

```
End Sub
```

```
Private Sub mnuBordeSen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles mnuBordeSen.Click
```

```
    mnuSencillo_Click(sender, e)
```

```
End Sub
```

D. ¿Cómo se activa el menú contextual?

Finalmente para lograr que el menú contextual aparezca al pulsar el botón derecho sobre el objeto Label, debemos indicar en la propiedad **ContextMenuStrip** el nombre del menú contextual creado. Es decir, el objeto **mnuContextual**.

Ya se encuentran en condiciones de probar su proyecto, recorran todo el menú para verificar lo programado y el menú contextual accionando el botón derecho del Mouse sobre el label de prueba.

 Guarde su proyecto como **Actividad 8**.

2) Cuadros de diálogo

Visual nos permite acceder a los cuadros de diálogo estándares de Windows, ellos son: el cuadro para seleccionar el color, el dialogo de Fuentes, el cuadro de Abrir y Guardar archivos, y demás.

Seguidamente describiremos estos cuadros de diálogo.

A. ColorDialog

Este control muestra el cuadro de dialogo del sistema operativo para la selección de colores.

Su propiedad más importante es la denominada **Color**, que contiene una estructura con el color seleccionado por el usuario aplicable a algunos de los elementos del formulario. Y el método más importante es **ShowDialog()** que permite mostrar el diálogo por pantalla.

Por ejemplo, si el color seleccionado queremos aplicarlo al fondo del formulario, así quedaría el código escrito:

ColorDialog1.ShowDialog() 'mostramos el dialogo Color
Me.BackColor=ColorDialog1.Color 'aplicamos el color al fondo del formulario
FontDialog

Este control exhibe el cuadro de dialogo de Fuentes del sistema para la selección de todo lo relacionado con la fuente salvo el color. Su propiedad más importante es la propiedad Font que contiene todo lo seleccionado por el usuario desde este cuadro de dialogo: tipo, estilo, tamaño y efectos de la fuente.

Por ejemplo, si la fuente seleccionada queremos aplicarla al texto de una etiqueta Label1, así quedaría el código escrito:

FontDialog1.ShowDialog()'mostramos el dialogo Fuente
Label1.Font=FontDialog1.Font

B. SaveFileDialog

Este control muestra el diálogo que conocemos como Guardar como.... Desde cualquier aplicación de Windows.

Este control no graba el archivo solo nos permite seleccionar la unidad, carpeta de destino y nombre del archivo que deseamos guardar.

Entre las propiedades más destacadas, citamos las siguientes:

- **Title**: contiene una cadena con el título que aparecerá en el cuadro de dialogo.
- **InitialDirectory**: carpeta predeterminada que se mostrará al abrir el cuadro de diálogo.
- **Filter**: permite determinar el tipo de archivo que se mostrará en el diálogo. El formato es el siguiente:

NombreArchivo(*.Extensión)|*.Extensión

Se pueden separar varias extensiones utilizando la barra vertical (|).

- **FileName**: nombre del archivo en el que se grabará.

C. OpenFileDialog

Este control muestra el cuadro de dialogo que conocemos como Abrir, en cual seleccionamos un archivo para luego poder realizar operaciones de lectura-escritura con él.

Comparte las mismas propiedades que las mencionadas en el cuadro de diálogo anterior.

3) RichTextBox (cuadro de texto enriquecido)

Es un control similar al cuadro de texto pero con características especiales de edición de texto y párrafos. Por ejemplo permite cambiar el color al texto seleccionado, abrir un archivo .txt o .rtf para mostrar su contenido como métodos para guardar archivos de estos tipos.

El método que nos permite abrir un archivo es el método **LoadFile**. Observen el siguiente fragmento de código.

```
Dim nombre as String 'defino la variable que contendrá el nombre del archivo
dlgAbrir.Title = "Abrir documento" 'dlgAbrir es un OpenFileDialog
dlgAbrir.InitialDirectory =
My.Computer.FileSystem.SpecialDirectories.MyDocuments
dlgAbrir.Filter =
"FORMATO Rtf(*.rtf)|*.rtf|Todos los archivos (*.*)|*.*"
nombre = dlgAbrir.FileName
RichTexto.LoadFile(nombre, RichTextBoxStreamType.RichText)
End If
```

En el método LoadFile hay que indicar no solo el nombre del archivo, sino también el tipo de archivo que se abrirá. En nuestro ejemplo un archivo .rtf.

El segundo método importante del RichTextBox que nos permite guardar su contenido en un archivo es el método **SaveFile**, similar al anterior. Preste atención en el siguiente código:

```
dlgGuardar.Title="Guardar documento" 'dlgGuardar es un SaveFileDialog
dlgGuardar.InitialDirectory = _ My.Computer.FileSystem.SpecialDirectories.MyDocuments
dlgGuardar.Filter = "Documentos (*.rtf)|*.rtf|Todos los archivos (*.*)|*.*"
nombre =
dlgGuardar.FileName
RichTexto.SaveFile(nombre, RichTextBoxStreamType.RichText)
```

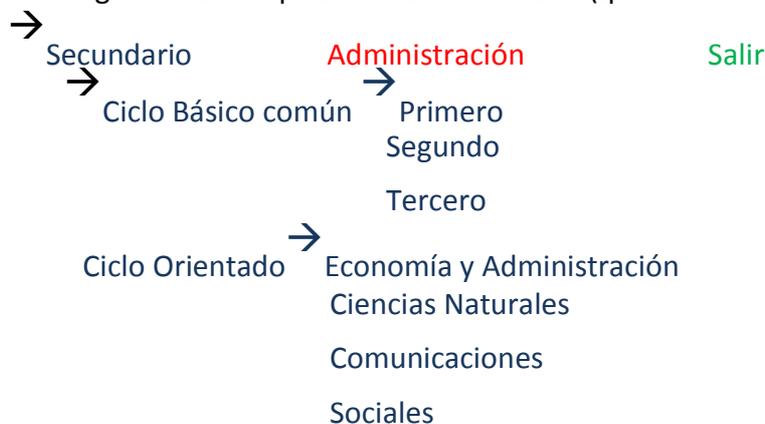
Al igual que en el metodo anterior se debe indicar nombre de archivo y tipo.



TAREA

Actividad Nº 10

Desarrollen el siguiente menú en un nuevo formulario, que titularán con el nombre de alguna institución educativa. Pueden complementarlo creando un menú contextual como atajo para acceder a algunas de las opciones de dicho menú (quedan a su criterio cuales incluirán).



 Guarden su actividad como **Actividad 10**.

Actividad N° 11

Consigna: Ingresar de 20 alumnos de una comisión, los siguientes datos dentro de múltiples vectores: apellido y nombre, sexo y nota final de cursada. Tomen en cuenta la explicación vertida en el Libro “Introducción a Vectores”.

Se pide:

- Calcular y mostrar un promedio general de todas las notas finales ingresadas.
- Calcular y mostrar cuantas mujeres superan el promedio general, obtenido en el punto anterior.
- Mostrar por pantallas todos los datos de los alumnos que no superen dicho promedio.

Ingreso de Calificaciones

Ingrese Nombre Sexo Nota Final 

Mostrar Resultados

1) Promedio general de todo el curso

2) Cantidad de mujeres que superan el promedio general

3) Alumnos que no superan el promedio general





 Guarden su actividad como **Actividad 11**.

Cierre de la clase

En esta clase se presentó el Editor de Menús de Visual Basic, utilizado para crear los menús personalizados estilo Windows para las aplicaciones. Aprendieron que es posible agregar o quitar marcas () a los elementos del menú, dividirlos en secciones mediante líneas de separación y ocultar sus elementos, estableciendo sus propiedades en tiempo de diseño.

Piensen que cualquier aplicación que trabaje bajo Windows debe contener un menú que permita a quien lo utilice navegar eficientemente por todas las opciones del programa, para poder ser tomado en cuenta como una aplicación profesional.

También aprendimos a utilizar los cuadros de diálogos propios del sistema operativo Windows que le servirá para agregar mayor funcionalidad a sus programas.

¡¡¡Les deseo a todos, que tengan una excelente semana!!

Murdaca Martin