

Ejercicios sobre Controles - I.

Curso 2010-2011

Práctica 1. Crear la primera aplicación

Abrir Visual Studio .NET y utilizar la Página de inicio

1. Abra Visual Studio .NET.
El panel Mi Perfil permite a los desarrolladores, que conocen versiones anteriores de Visual Basic o Microsoft Visual C++®, personalizar el entorno de desarrollo para que sea similar a dichas versiones. Para esta demostración, no cambie la configuración predeterminada. Este panel aparece cuando Visual Studio .NET se abre por primera vez y, desde ese momento, estará localizado en la Página de inicio.
2. Haga clic en **Comunidad Online**.
Este panel proporciona fácil acceso a grupos de noticias relacionados con el desarrollo .NET.
3. Haga clic en **Búsqueda Online**.
Este panel proporciona fácil acceso a la biblioteca en línea de Microsoft MSDN®.
4. Haga clic en **Inicio**.


Crear un nuevo proyecto

1. En la pestaña **Proyectos**, haga clic en **Nuevo Proyecto**.
2. En el cuadro de diálogo **Nuevo Proyecto**, en el panel **Plantillas**, haga clic en **Aplicación para Windows**.
3. En el cuadro **Nombre**, escriba **ApplicationToExplore**.
4. Para establecer la ubicación, haga clic en **Examinar** y vaya a la carpeta donde quiera crear el proyecto, haga clic en **Abrir** y, a continuación, haga clic en **Aceptar**.
Nota: Una plantilla de Aplicación para Windows en Visual Basic .NET proporciona un único formulario Windows. El formulario se denomina Form1.vb, y se abre en la vista Diseño.

Examinar el Explorador de soluciones

1. En el Explorador de soluciones, haga clic en **Form1.vb**, si no está seleccionado.
Observe que el archivo Form1.vb ha sido proporcionado por la plantilla de Aplicación para Windows utilizada para crear el proyecto.
2. Haga clic con el botón derecho en **Solución 'ApplicationToExplore' (1 proyecto)**.
Observe que puede añadir otros proyectos a la solución además de poder generar y depurar la solución desde este menú de acceso directo.
3. Haga clic con el botón derecho en **ApplicationToExplore** debajo del nodo de Solución.
Observe que el menú de acceso directo del proyecto incluye opciones para añadir referencias y otros archivos al proyecto.

Cambiar las propiedades del formulario

1. Para hacer que el formulario sea rectangular, arrastre el ajuste de tamaño en la parte inferior derecha.
2. En la ventana Propiedades, haga clic en **BackgroundImage** y, a continuación, haga clic en el botón de puntos suspensivos .
3. En el cuadro de diálogo **Abrir**, vaya a la carpeta en la que se encuentre el fichero **Blue hills.jpg** (este fichero se puede encontrar en la red), selecciónelo, y a continuación haga clic en **Abrir**.

Añadir botones al formulario

1. Haga clic en el control **Button** del Cuadro de herramientas, y arrástrelo al formulario.
2. Arrastre un segundo botón al formulario.

Añadir código a un botón


1. Haga doble clic en **Button1** en el formulario.
Observe que se abre el Editor de código. La pestaña de la ventana se denomina Form1.vb*. El símbolo * que aparece al final del nombre significa que los últimos cambios realizados al control no se han guardado en el disco.
2. En el Editor de código, entre Private Sub Button1_Click y End Sub (donde debería estar ubicado el cursor), escriba el siguiente código:

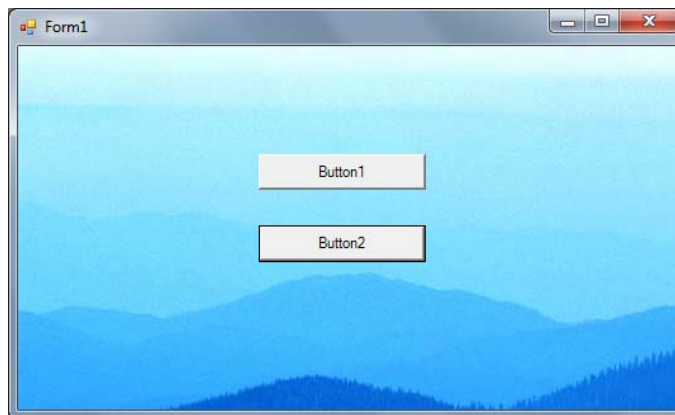
```
MessageBox.Show("Ready....Get Set.... Write Code!!!")
```

3. Abra la vista de Diseño para Form1.vb.
4. Haga doble clic en **Button2** en el formulario.
5. En el Editor de código, entre Private Sub Button2_Click y End Sub (donde debería estar ubicado el cursor), escriba el siguiente código:

```
MessageBox.Show("Web....Windows...." & _  
"Visual Studio .NET does it all!")
```


Ejecutar y probar la aplicación

1. En la barra de herramientas estándar, haga clic en el botón **Iniciar** . Esto también se puede hacer mediante la tecla **F5**.
La aplicación tendrá la siguiente apariencia:



2. En el cuadro de diálogo **Form1**, haga clic en **Button1**.
3. En el cuadro de mensaje, haga clic en **OK**.
4. En el cuadro de diálogo **Form1**, haga clic en **Button2**.
5. En el cuadro de mensaje, haga clic en **OK**.
6. Cierre **Form1**.

Utilizar la funcionalidad de ocultación automática (Auto Hide)

1. En la barra de título de la ventana de Resultados, haga clic en el icono del pin .
Observe cómo desaparece la ventana de Resultados cuando el cursor se aleja de la ventana y se abre de nuevo cuando el cursor se posiciona sobre la pestaña **Resultados**.

Desacoplar una ventana

1. Haga clic en el icono del pin de la barra de título del Cuadro de herramientas.

2. Haga clic en la barra de título del Cuadro de herramientas y arrastre la ventana al centro de la pantalla.

Acoplar una ventana

1. Haga clic en la barra de título del Cuadro de herramientas y arrastre la ventana al lado izquierdo de la ventana de la aplicación de Visual Studio.NET hasta que el cursor esté casi en el borde izquierdo de la pantalla.
2. Haga clic en el icono del pin de la barra de título del Cuadro de herramientas para habilitar de nuevo la funcionalidad de ocultación automática (Auto Hide).

Guardar un proyecto

1. En la barra de herramientas estándar, haga clic en el botón **Guardar todo**.
La ubicación de la solución se establece cuando se crea la solución.

Cerrar una solución

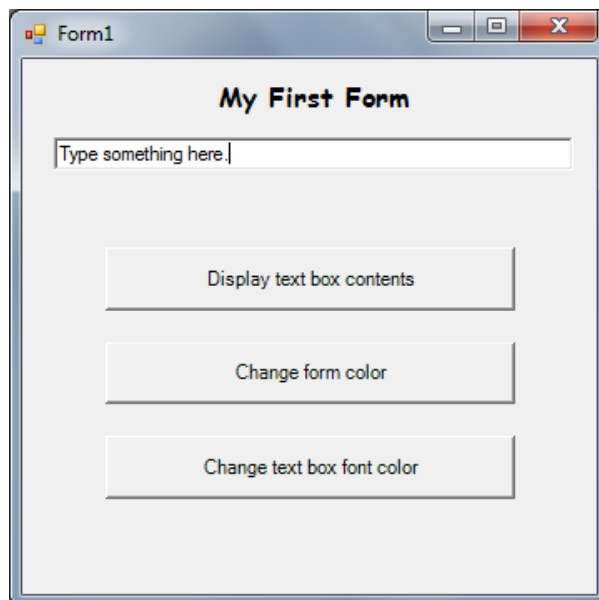
1. En el menú **Archivo**, haga clic en **Cerrar Solución**.
La pestaña **Proyectos** de la Página de inicio mostrará **ApplicationToExplore** en la lista de proyectos recientes cuando la pantalla se actualice o la próxima vez que se abra Visual Studio .NET.

Salir de Visual Studio .NET

1. En el menú **Archivo**, haga clic en **Salir**.

Práctica 2. Añadir controles básicos a un formulario

En esta práctica se creará una aplicación sencilla basada en Windows y se añadirán controles a un formulario. También se creará una versión *Release* de la aplicación para poder ejecutarla fuera del entorno de desarrollo. El aspecto de la aplicación es el siguiente:



Funcionalidad:

1. Cuando el usuario hace clic en **Display text box contents**, aparece un cuadro de mensaje mostrando el contenido del cuadro de texto.
2. Cuando el usuario hace clic en **Change form color**, cambia el color de fondo del formulario.

Por regla general, las directrices de accesibilidad sugieren que los desarrolladores no deberían cambiar el color de fondo de los formularios (Por ello la gran mayoría de las aplicaciones tienen una apariencia similar y los controles tienen idénticas funcionalidades). Si el color de fondo no se cambia, el formulario incorpora la configuración de Windows. Sin embargo, en algunos casos es posible que necesite cambiar los colores de fondo debido a los requisitos de diseño.

3. Cuando el usuario hace clic en **Change text box font color**, cambiará el color del texto del cuadro de texto.


Iniciar una nueva aplicación en Visual Basic .NET

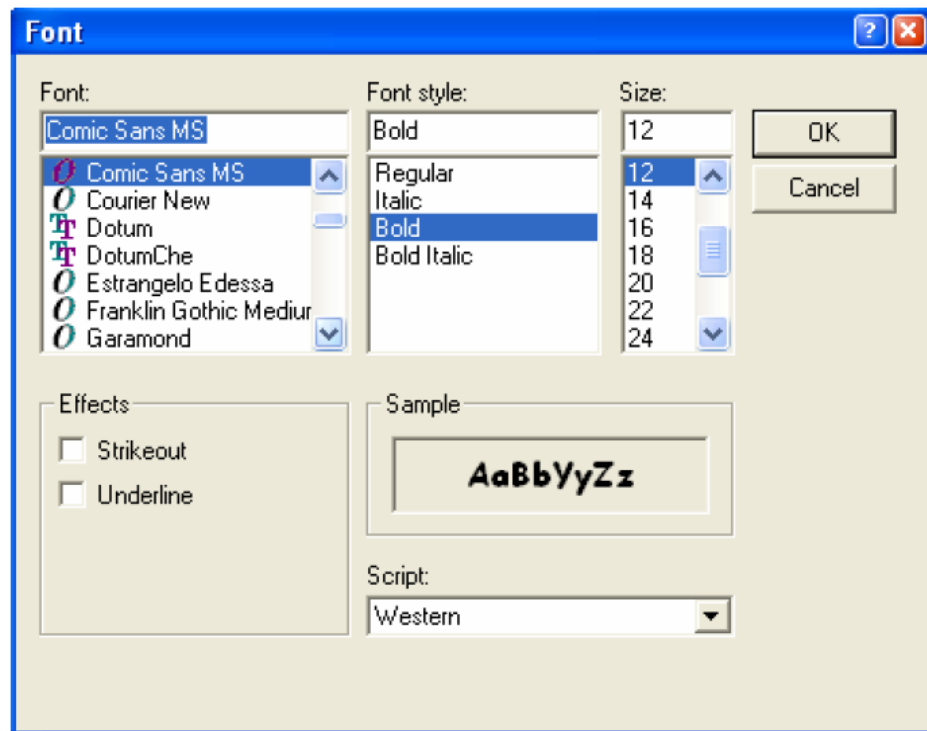
1. Abra Visual Studio .NET.
2. En el menú **Archivo**, seleccione **Nuevo** y haga clic en **Proyecto**.
3. En el cuadro de diálogo **Nuevo Proyecto**, haga clic en **Proyectos de Visual Basic** y, a continuación, haga clic en **Aplicación para Windows**. Establezca **MyFirstProject** como nombre del proyecto, seleccione la ubicación deseada y, a continuación, haga clic en **Aceptar**.

Añadir controles al formulario

1. Añada un control **Label** al formulario.
Para añadir un control en una ubicación específica del formulario, haga clic sobre el control deseado y, a continuación, haga clic en la ubicación elegida del formulario. El control se añadirá a esa ubicación.
2. Añada un control **TextBox** al formulario.
3. Añada tres controles **Button** al formulario.
4. Cambie de lugar y de tamaño el formulario y los controles hasta que el formulario tenga un aspecto similar a la imagen que se muestra al inicio de esta práctica.
5. Guarde su trabajo.

Establezca propiedades para los controles

1. Haga clic en **Label1** en el formulario para ver sus propiedades en la ventana Propiedades.
Si la ventana Propiedades no está visible, en el menú **Ver**, haga clic en **Ventana Propiedades**.
2. En la ventana Propiedades, haga clic en la propiedad **Font** y, a continuación, haga clic en el botón de puntos suspensivos  para mostrar el cuadro de diálogo **Font**. Cambie la configuración de la fuente como se muestra en la siguiente figura y, a continuación, haga clic en **Aceptar**.



3. Establezca las propiedades para la etiqueta y el cuadro de texto como se muestra en la siguiente tabla:

Nombre del control	Propiedad	Nuevo valor
Label1	Text	My First Form
Label1	TextAlign	MiddleCenter
TextBox1	Text	Escriba algo aquí

4. Establezca la propiedad **Text** para los tres botones como se muestra en la siguiente tabla:

Nombre del control	Propiedad	Nuevo valor
Button1	Text	Display text box contents
Button2	Text	Change form color
Button3	Text	Change text box font color

5. Guarde su trabajo.

Nota: En los tres procedimientos siguientes, utilizará el Editor de código de Form1.vb para añadir varias líneas de código a su aplicación para que pueda responder a la entrada del usuario.

Añadir código al manejador de eventos Click del botón Display text box contents

1. Haga doble clic en el botón **Display text box contents** del formulario. Se abrirá el Editor de código de Form1.vb con el siguiente código ya insertado:

```
Private Sub Button1_Haga clic en ...
End Sub
```

El formulario tiene dos vistas: la vista de diseño (Form1.vb[Diseño]) y la vista de código (Form1.vb). Puede moverse entre una y otra haciendo clic en las pestañas de la parte superior del formulario.

2. Escriba la siguiente línea de código en el gestor de eventos de **Button1_Click**:

```
MessageBox.Show(TextBox1.Text)
```

Añadir código para cambiar el color del botón y el fondo del formulario

1. Cambie a la vista de Diseño del formulario.
2. Haga doble clic en el botón **Change form color** para abrir el Editor de código.
3. Escriba la siguiente línea de código en el gestor de eventos de **Button2_Click**:

```
Me.BackColor = Color.CadetBlue
```

Añadir código para cambiar el color de la fuente del cuadro de texto

1. Cambie a la vista de Diseño del formulario.
2. Haga doble clic en el botón **Change text box font color** del formulario para abrir el Editor de código.
3. Escriba la siguiente línea de código en el gestor de eventos **Button3_Click**:

```
TextBox1.ForeColor = Color.Red
```

Ejecute y pruebe su aplicación en el entorno de diseño

1. Haga clic en el botón **Iniciar** de la barra de herramientas estándar.
 1. Haga clic en todos los botones del formulario que se está ejecutando para asegurarse de que funcionan correctamente.
 2. Cambie el texto del cuadro de texto y, a continuación, haga clic en **Display text box contents**.
 3. Salga de la aplicación.

Guarde y genere un archivo ejecutable para su aplicación

1. Para guardar su solución, haga clic en el botón **Guardar todo** de la barra de herramientas estándar.
2. En la barra de herramientas estándar, en la lista **Configuraciones de Solución**, cambie la configuración de la solución de **Debug** a **Release**.
3. En el menú **Generar**, haga clic en **Generar solución** para crear una versión *Release* de su aplicación.
4. Salga de Visual Studio .NET.
5. En Windows Explorer, localice su archivo ejecutable en la carpeta `..\MyFirstProject\obj\Release`, dentro de la localización especificada al crear el proyecto.
6. Haga doble clic en el archivo ejecutable para ejecutarlo, y pruebe todas las funcionalidades fuera del IDE. A continuación, cierre la aplicación.

Práctica 3. Crear una aplicación con varios formularios

En esta práctica se creará una aplicación sencilla basada en Windows y se añadirán controles al formulario principal para organizarlos adecuadamente. También se añadirán otros formularios con diferentes funcionalidades.

Iniciar una nueva aplicación en Visual Basic .NET

1. Abra Visual Studio .NET.
2. Añada un nuevo proyecto de tipo **Windows Form**.

3. Asígnele un nombre al proyecto y guárdelo en una carpeta del ordenador. Recuerde guardar los cambios cada cierto tiempo.
4. Haga clic en el formulario para seleccionarlo y establezca las siguientes propiedades:

Propiedad	Nuevo Valor
Name	frmPrincipal
Text	Práctica 1-3

Añadir controles al formulario principal

1. Añada un botón al formulario y sitúelo lo más cercano posible al borde superior izquierdo del formulario. Establezca las siguientes propiedades:

Propiedad	Nuevo Valor
Name	btnCalendario
Text	Calendario

2. Añada otro botón al formulario y sitúelo en la misma posición horizontal que ocupa actualmente pero lo más cercano posible al borde derecho del formulario. Establezca las siguientes propiedades:

Propiedad	Nuevo Valor
Name	btnHora
Text	Hora

3. Añada una caja de texto (**Textbox**) de un tamaño tal que sus bordes izquierdo y derecho no coincidan con los respectivos bordes del formulario contenedor. Establezca las siguientes propiedades:

Propiedad	Nuevo Valor
Name	txtFecha
Font	Times New Roman, Normal, 12

4. Añada una etiqueta (**Label**) y sitúela debajo de la caja de texto. Establezca las siguientes propiedades:

Propiedad	Nuevo Valor
AutoSize	False
Name	lblDiaSemana
Font	Arial, Normal, 14
TextAlign	MiddleCenter (Cuadro central)

5. Añada otro botón al formulario y sitúelo lo más cercano posible al borde inferior del formulario. Establezca las siguientes propiedades:

Propiedad	Nuevo Valor
Name	btnFin
Text	Finalizar

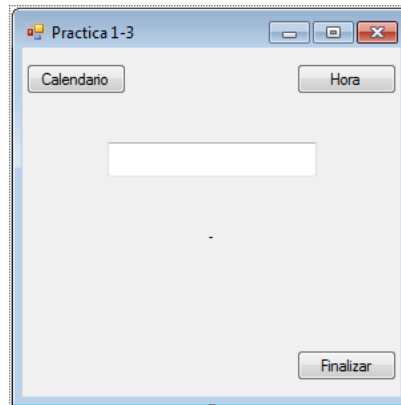
Alinear y situar los controles.

El menú **Formato** ofrece numerosas opciones para organizar los controles en función de la posición de uno de ellos o en función de las dimensiones del contenedor. Cuando se utilizan las opciones del menú **Formato** para organizar varios controles, es conveniente seleccionarlos de forma que el primer control seleccionado sea el control primario respecto al cual se situará el resto. Las marcas de selección del control primario son cuadros de color blanco, mientras que las marcas de selección del resto de controles son cuadros de color negro.

Para seleccionar varios controles se hace clic en el primero y a continuación se hace clic en todos los demás, manteniendo la tecla **CTRL** presionada.

1. Seleccione primero el botón **btnCalendario** y luego el botón **btnHora**.
2. Active la opción **Formato** del menú principal.
3. Seleccione la opción **Alinear** → **Puntos Medios**. Con esta acción ambos controles quedarán centrados con relación al punto medio horizontal del botón **btnCalendario**, es decir, situados en la misma línea horizontal.
4. Si los controles no continúan seleccionados, hágalo como se indica en el apartado a. Seleccione la opción **Alinear** → **Centro**. Con esta acción ambos controles quedarán a la misma distancia del borde izquierdo y del borde derecho del formulario contenedor.
5. Seleccione la caja de texto.
6. Active la opción **Formato** del menú principal y seleccione **Centrar en el formulario** → **Horizontalmente**.
7. Seleccione la caja de texto y a continuación seleccione la etiqueta. Active la opción **Formato** del menú principal y seleccione **Alinear** → **Puntos Medios**.
8. Seleccione primero el botón **btnHora** y luego el botón **btnFin**.
9. Active la opción **Formato** del menú principal y seleccione **Alinear** → **Lados derechos**.

El formulario principal debe tener este aspecto:



Garantizar que se mantenga la posición relativa de los controles en el formulario.

Ejecute la aplicación y observe lo que sucede con la apariencia cuando se modifica el tamaño del formulario.

El tamaño y posición de los controles del formulario debería ajustarse correctamente si el tamaño del formulario contenedor se modifica en tiempo de ejecución. Para realizar el ajuste de posición se utiliza la propiedad **Anchor** de los controles.

Los controles están ajustados hacia arriba y hacia la izquierda (**Top, Left**) de forma predeterminada. Para borrar o seleccionar un lado de ajuste del control se hace clic en la barra correspondiente a ese lado que se muestra al activar esta propiedad.

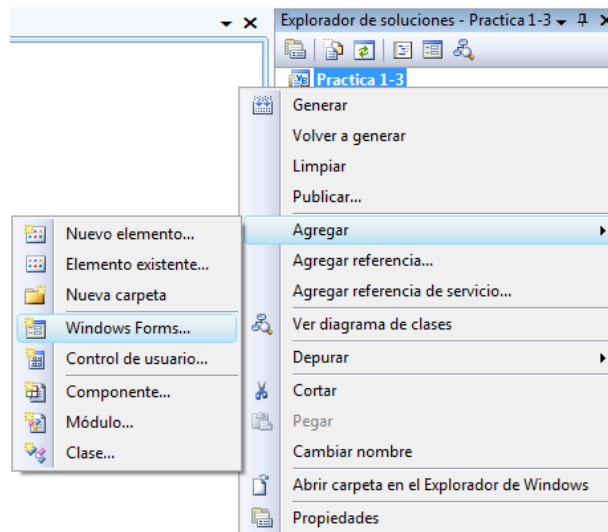
1. Seleccione el botón **btnCalendario**.
2. Active la propiedad **Anchor** en la ventana de propiedades.
3. Compruebe que el ajuste está seleccionado a los valores predeterminados (**Top, Left**). En caso contrario active el editor de los valores de la propiedad en la columna de la derecha. Borre los valores establecidos y seleccione el rectángulo de la parte superior y el de la parte izquierda.
4. Seleccione uno a uno el resto de los controles del formulario y establezca los valores de la propiedad **Anchor** de acuerdo a la tabla siguiente:

Control	Propiedad Anchor
btnHora	Top, Right (Arriba, Derecha)
btnFin	Botton, Right (Abajo, Derecha)
txtFecha	None (Ninguno)
lblDiaSemana	None (Ninguno)

5. Ejecute la aplicación y modifique el tamaño de la ventana de la aplicación. Observe el efecto que han ocasionado los cambios realizados en la propiedad Anchor.

Añadir nuevos formularios al proyecto.

1. Haga clic con el botón derecho del ratón encima del nombre de la solución que aparece en la ventana del **Explorador de Soluciones**.
2. Seleccione la opción **Agregar** → **Windows Forms**.



3. Asígnele el nombre **Login.vb** en la ventana que se muestra y haga clic en el botón **Aceptar**.
4. Establezca los valores a las propiedades que se indican en la tabla a continuación. **Observe en el panel de descripción, en la parte inferior de la ventana de propiedades, el significado de cada una de las propiedades seleccionadas.**

Propiedad	Nuevo Valor
Name	frmLogin
ControlBox	False
FormBorderStyle	Fixed3D
MaximizeBox	False
MinimizeBox	False
Size	322; 210
StartPosition	CenterScreen
Text	Autenticación
TopMost	True

5. Agregar al formulario dos etiquetas, dos cajas de texto y dos botones.
6. Establecer las propiedades de los controles, de acuerdo a la siguiente tabla:

Control	Propiedad	Nuevo Valor
Label1	Name	lblNombre
	Location	64; 31
	Size	63; 14
	Text	Usuario
Label2	Name	lblClave
	Location	64; 71
	Size	64; 14
	Text	Clave
Textbox1	Name	txtNombre
	Location	128; 29
	Size	120; 20
	Text	<i>Dejar en blanco</i>
Textbox2	Name	txtClave
	Location	128; 64
	Size	120; 20
	PasswordChar	*
	Text	<i>Dejar en blanco</i>
Button1	Name	btnEntrar
	Location	67; 116
	Size	75; 30
	Text	&Entrar
Button2	Name	btnFin
	Location	171; 116
	Size	75; 30
	Text	&Salir

7. Ajustar la posición de los controles para que el aspecto de la ventana sea similar a la que se muestra a continuación:

8. Completar las propiedades del formulario **frmLogin** según la siguiente tabla:

Propiedad	Nuevo Valor
AcceptButton	btnEntrar
CancelButton	btnSalir

9. Agregue otro formulario a la aplicación y póngale por nombre **Hora.vb** cuando aparezca la ventana correspondiente.
10. Asígnele las propiedades que se indican en la tabla siguiente:

Propiedad	Nuevo Valor
Name	frmHora
MaximizeBox	False
MinimizeBox	False
Size	315; 165
Text	Hora

11. Agregue dos etiquetas al formulario y sitúelas en el centro del mismo, en líneas independientes. Asígnele las siguientes propiedades:

Control	Propiedad	Nuevo Valor
Label1 (superior)	Name	lblTitulo
	Font	Arial, Normal, 12
	Text	Hora Actual
Label2	AutoSize	False
	Name	lblHora
	Font	Courier New; Bold, 20
	Size	210; 35
	TextAlign	MiddleCenter
	Text	-

12. Agregue un control Timer al formulario. Asígnele las propiedades que se indican en la tabla siguiente:

Propiedad	Nuevo Valor
Name	tmrHora
Interval	1000

Añadir funcionalidad a los controles mediante manejadores de eventos

- En el editor de la herramienta de desarrollo, active la pestaña correspondiente a la ventana de diseño del formulario principal de la aplicación (**Form1.vb[Diseño]**).
- Haga doble click en el botón **Calendario** para añadir un manejador para el evento **Click**.
- Inserte las líneas de código siguientes cuando se haya generado el procedimiento:

```
Dim d As DateTime


d = Convert.ToDateTime(txtFecha.Text)
lblDiaSemana.Text = d.ToString("dddd")
```

- Siguiendo el procedimiento indicado en el apartado 1, agregue un manejador para el evento **Click** del botón **Hora** con el siguiente código:

```
frmHora.Show()
```

- Siguiendo el procedimiento indicado en el apartado 1, agregue un manejador para el evento **Click** del botón **Finalizar** con el siguiente código:

```
Me.Close()
```

- Haga clic en el área del formulario para seleccionarlo.
- En la ventana de propiedades, seleccione el botón , que muestra el listado de los eventos.
- Haga doble clic en el evento **Load** para insertar un procedimiento manejador para el mismo.

9. Inserte el siguiente código:

```
Me.Enabled = False  
frmLogin.Focus()  
frmLogin.Show()
```

Nota:

Llegados a este punto, es preciso que se conozca bien el mecanismo para añadir manejadores de eventos a los controles. A partir de ahora se obviarán los pasos para realizar esta acción, con el objetivo de incidir en otros aspectos más importantes.

10. En el formulario **Login**, añadir un procedimiento manejador del evento **Click** para el botón **Entrar**. El código a insertar es el siguiente:

```
If txtNombre.Text.CompareTo("alumno") = 0 AndAlso _  
    txtClave.Text.CompareTo("montecastelo") = 0 Then  
    frmPrincipal.Enabled = True  
    Me.Hide()  
Else  
    MessageBox.Show("Nombre de usuario o clave incorrecta")  
    frmPrincipal.Close()  
End If
```

11. En el formulario **Login**, añadir un procedimiento manejador del evento **Click** para el botón **Salir**. El código a insertar es el siguiente:

```
MessageBox.Show("Debe especificar un nombre de usuario _  
                y contraseña")  
frmPrincipal.Close()
```

12. En el formulario **Hora**, añadir un procedimiento manejador del evento **Tick** para el temporizador **tmrHora**. El código a insertar es el siguiente:

```
lblHora.Text = Date.Now.ToLongTimeString
```

13. Añadir un procedimiento manejador del evento **Load** para el formulario **Hora**. El código a insertar es el siguiente:

```
tmrHora.Start()
```

14. Añadir un procedimiento manejador del evento **FormClosing** para el formulario **Hora**. El código a insertar es el siguiente:

```
tmrHora.Stop()
```

Comprobar el funcionamiento de la aplicación

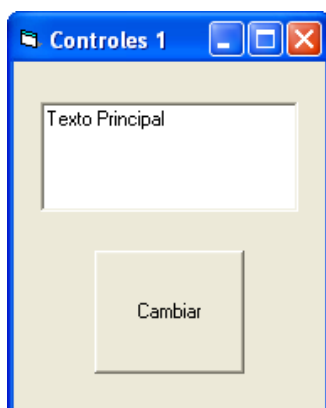
1. Una vez asegurado que en la ventana **Lista de Errores** (zona inferior) no hay ningún mensaje de error, puede ejecutar la aplicación para verificar su funcionalidad.
2. Utilice “alumno” y “montecastelo” como nombre de usuario y clave, respectivamente, al inicio de la aplicación.
3. Haga los cambios que estime oportunos para perfeccionar la aplicación o comprender mejor su funcionamiento.

Ejercicios

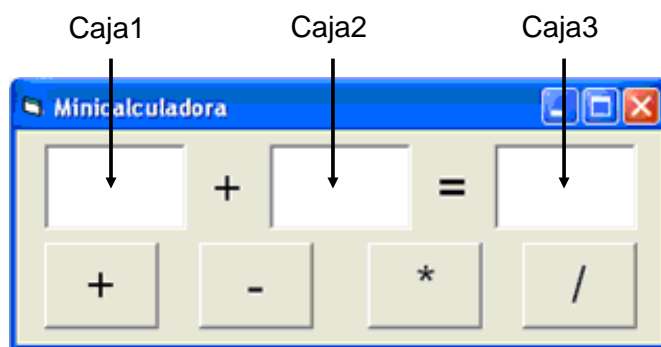
Estos ejercicios tienen por objetivo mejorar las habilidades en el diseño de aplicaciones para Windows que utilizan controles sencillos como botones, etiquetas y cajas de textos.

1. Desarrollar un programa que al pulsar un botón cambie el contenido de una caja de texto. El texto presentado alternará entre las cadenas “Texto Principal” y “Texto Secundario”. Al inicio en la ventana de texto se mostrará la primera cadena.

La ventana tendrá el siguiente aspecto:



2. Añadir las instrucciones necesarias para que junto con el cambio del texto de la caja, también de forma alterna, se modifique el color de fondo y el de los caracteres de dicho texto cada vez que se pulse el botón.
Las propiedades a modificar son: *ForeColor* y *BackColor*
3. Añadir otro botón al formulario para abandonar la aplicación. Mediante la instrucción *End*, se finaliza el programa.
4. Desarrollar una aplicación con la apariencia mostrada, en la que se implementa una calculadora elemental.



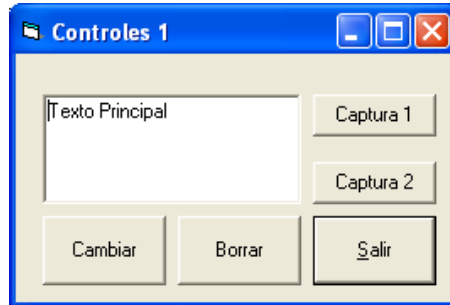
En las cajas de texto 1 y 2 se escriben números reales que actúan como operandos de los operadores mostrados en los botones. En la caja 3 se muestra el resultado de la operación. La apariencia inicial de la aplicación es la mostrada en la figura anterior.

El usuario introduce los datos y a continuación presiona uno de los botones que constituyen la operación a realizar. Además de mostrar el resultado, entre las cajas 1 y 2 aparece el símbolo de la operación realizada, para lo cual se utiliza un control *Label* (etiqueta).

5. Tal como está diseñada la aplicación, una vez que se realiza una operación matemática y se muestra el resultado, el usuario puede cambiarlo ya que la caja de texto

correspondiente está habilitada para escritura. Añadir a la aplicación la posibilidad de que NO se pueda cambiar el valor que aparece en la caja de texto del resultado.

6. Añadir a la aplicación la posibilidad de evitar un error de ejecución debido a la división por cero. Para ello debe lanzarse un mensaje de error y borrar el contenido de la caja de texto 2. Para lanzar el mensaje de error se utiliza el control *MessageBox* (buscar documentación en MSDN de la Web)
7. Desarrollar una aplicación con la apariencia inicial mostrada en la siguiente figura:



Los botones “Captura 1” y “Captura 2” se utilizan para almacenar dos textos, respectivamente, que se escriben en la caja de texto. El botón “Cambiar” muestra alternativamente, cada vez que se accione, los textos almacenados. El botón “Borrar” elimina el contenido visible de la caja de texto. El botón “Salir” permite abandonar la aplicación.

8. Realizar un programa con interfaz gráfica de usuario que permita leer un dato de tipo real. El dato que solo podrá tener valores entre 0 y 10,0 (ambos incluidos) significa la nota de un estudiante. Al pulsar un botón, se debe mostrar un texto en el formulario de un color determinado, que depende de las especificaciones establecidas en la siguiente tabla:

Valor	Texto	Color del texto
Entre 0 y 5,0 (ambos incluidos)	Suspense	Rojo
Entre 5 y 6,5 (incluido)	Aprobado	Amarillo
Entre 6,5 y 8,5 (incluido)	Notable	Azul cielo
Entre 8,5 y 10	Sobresaliente	Verde
10,0	Matrícula de honor	Azul

9. Realizar un programa con Interfaz Gráfica de Usuario que simule las tiradas de un dado. En la ventana de la aplicación debe aparecer el resultado de la tirada, la cantidad de tiradas que se han realizado y la frecuencia relativa de cada uno de los valores posibles. La simulación de la tirada debe realizarse con un botón y el resultado será un número aleatorio entero entre 1 y 6 (ambos incluidos).

Sugerencia: Para darle mejor apariencia a la aplicación se puede mostrar una imagen con la cara del dado correspondiente al resultado de la tirada. Para ello se utilizan varios controles *PictureBox*, cada uno de ellos con la imagen de una de las caras del dado. El procedimiento es hacer visible el control que contiene la imagen de la cara correspondiente al resultado y ocultar todas las demás. Se deben colocar los controles *PictureBox* uno encima del otro para que la imagen visible siempre esté en el mismo lugar.

Se proporcionan las imágenes de las caras del dado en los archivos *dado_uno.jpg* al *dado_seis.jpg*.

Propiedades del *PictureBox* a modificar:

Image (Nombre de la imagen a mostrar)

SizeMode = *StretchMode* (La imagen se adapta al tamaño del control)

10. Repita el ejercicio anterior utilizando botones en lugar de PictureBox como contenedores de las imágenes de las caras del dado. Valore la utilización de la propiedad **ImageList** del botón. debe tenerse en cuenta que para acceder a la funcionalidad de esta propiedad hay que insertar un control **ImageList** en la aplicación.