

PROVINCIA DE BUENOS AIRES
DIRECCIÓN GENERAL DE CULTURA Y EDUCACIÓN
DIRECCIÓN DE EDUCACIÓN SUPERIOR
INSTITUTO SUPERIOR DE FORMACIÓN TÉCNICA N° 93



“SEMINARIO DE PROGRAMACIÓN”

Cuadernillo 2023

CARRERA: Tecnicatura Superior en Análisis, Desarrollo y Programación de Aplicaciones.

CURSO: Segundo año

CANTIDAD DE HORAS: 64 módulos

PROFESORES: Landini Silvana – Pacífico Ezequiel

INTROODUCCION AL VISUAL C#

Para comenzar con Seminario de Programación, haremos un primer acercamiento a los contenidos que abordaremos con mayor profundidad en las siguientes clases.

Debido a que la cátedra es mayormente práctica, comenzaremos con las actividades de paso a paso, forma en la cual se seguirá a cada una de las guías propuestas en este cuadernillo.

Guía de laboratorio - Capítulo I

Contenidos:

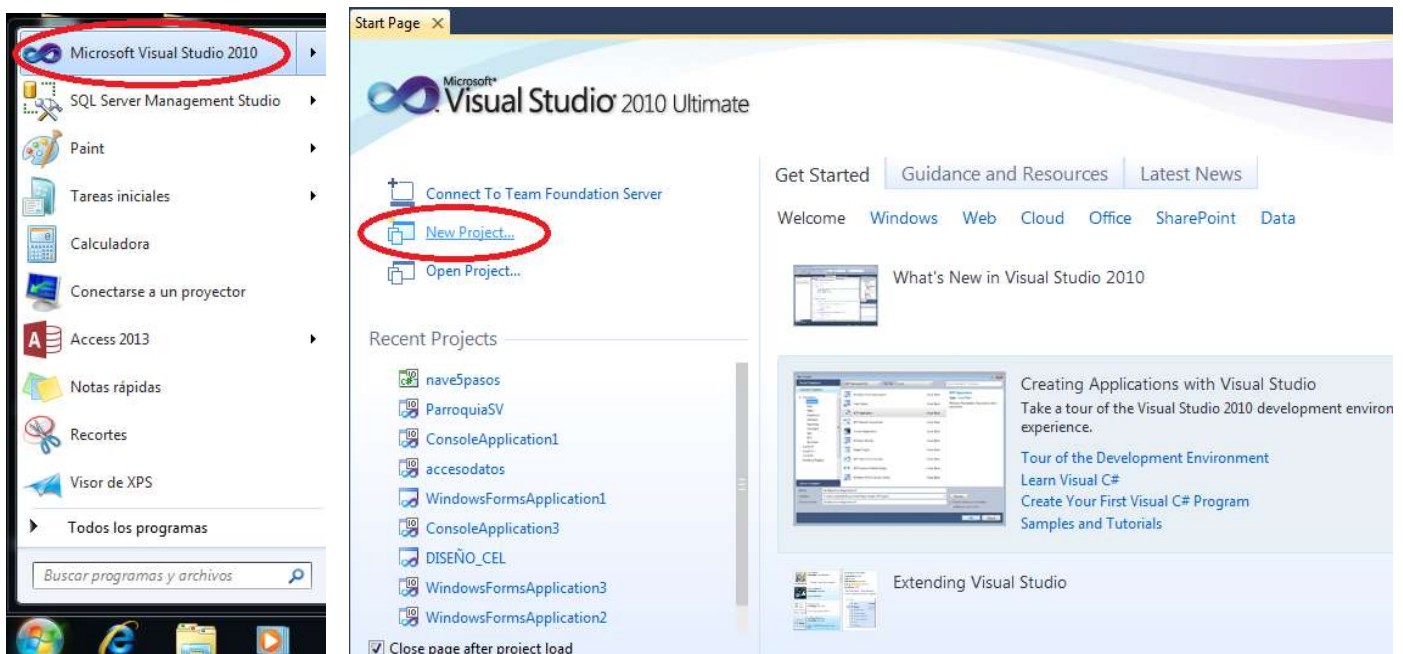
- ✓ Crear un proyecto con C#;
- ✓ Manipular controles y sus propiedades en tiempo de diseño;
- ✓ Ejecutar un programa;
- ✓ Manejar un evento de clic de botón;
- ✓ Mostrar un cuadro de mensaje;
- ✓ Colocar texto en una etiqueta en tiempo de ejecución;

Fuente principal: B. Douglas & P. Mark (2010). "C# para estudiantes". Pearson Educación. DF México.

Creando un nuevo proyecto

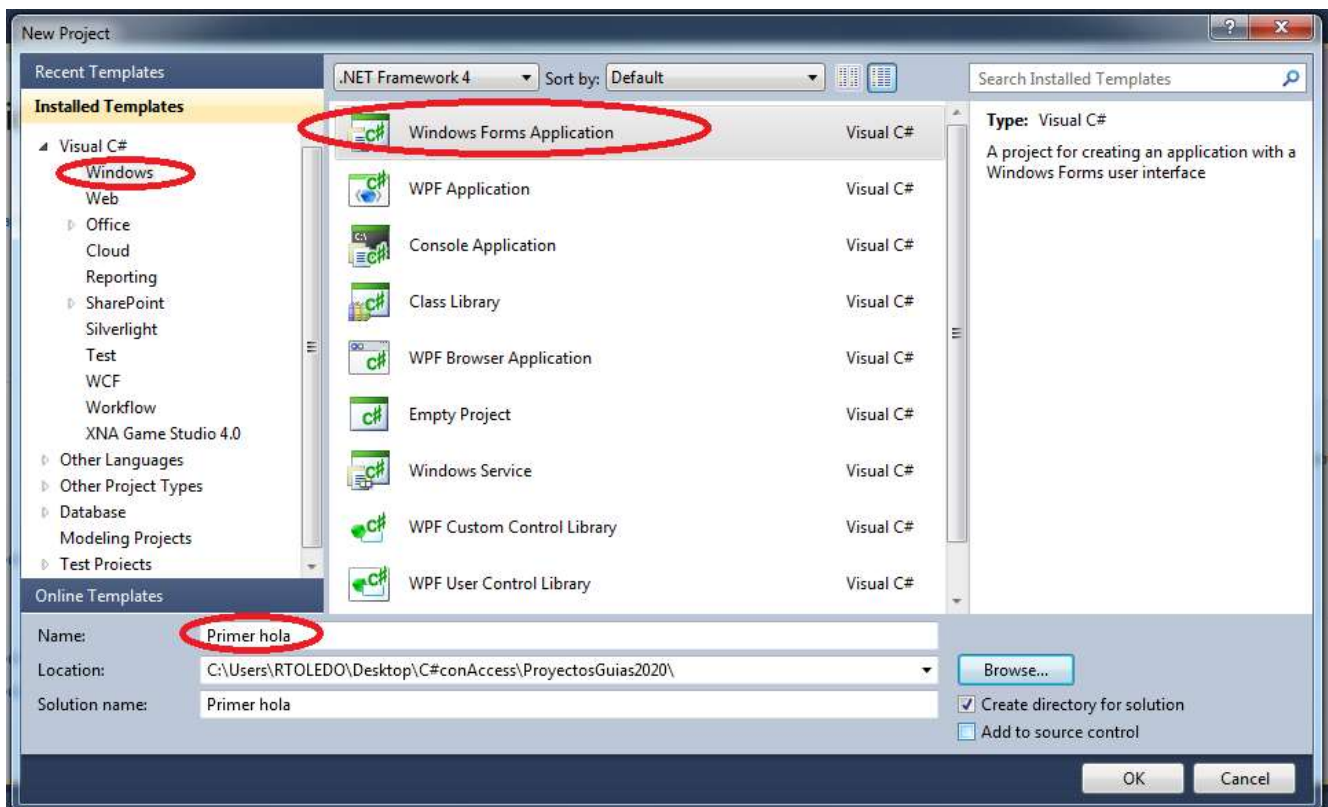
En primer lugar seleccionamos de nuestros programas, el IDE (Integrated Development Environment) o mejor conocido como entorno de desarrollo completo de Microsoft Visual Studio (y la versión).

A continuación aparecerá la **Página de inicio**, como se muestra en la Figura:

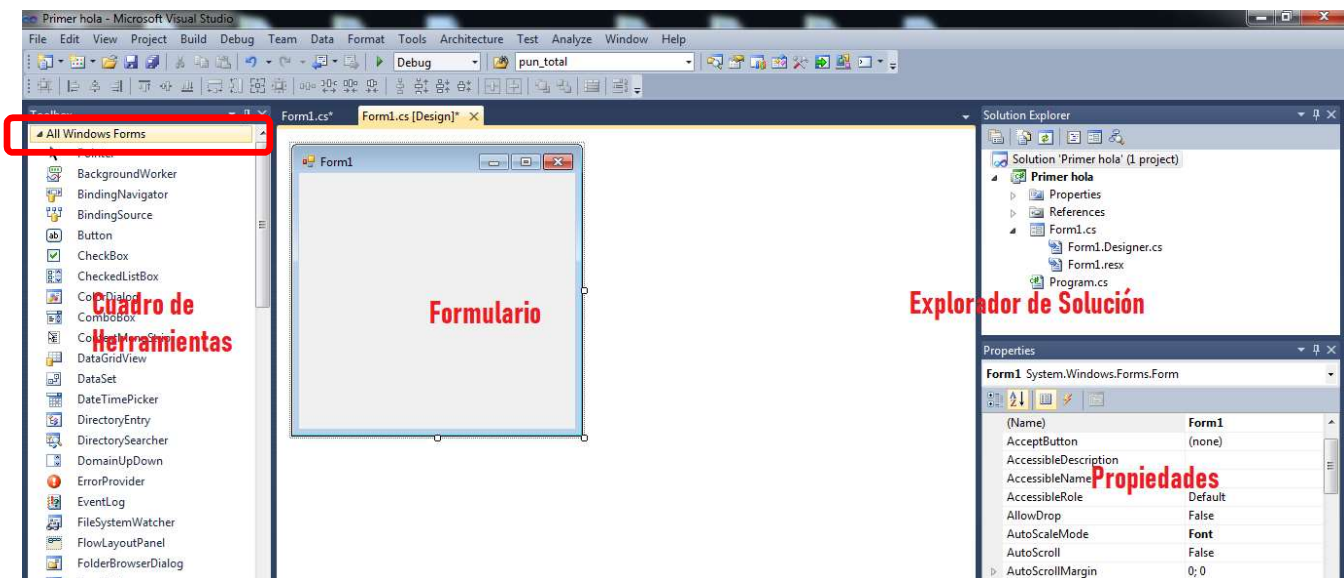


Hacer clic en el vínculo **Nuevo proyecto** y en la siguiente ventana asegurarse de que esté seleccionado **Visual C#**: **Windows** y la plantilla **Aplicación de Windows Forms**.

Elegir un nombre para el proyecto; el mismo se convertirá también en la identificación de una carpeta. En este caso utilizamos el nombre "Primer hola". Hacer clic en OK.



A continuación aparecerá un área de diseño similar (aunque no necesariamente idéntica) a la siguiente:

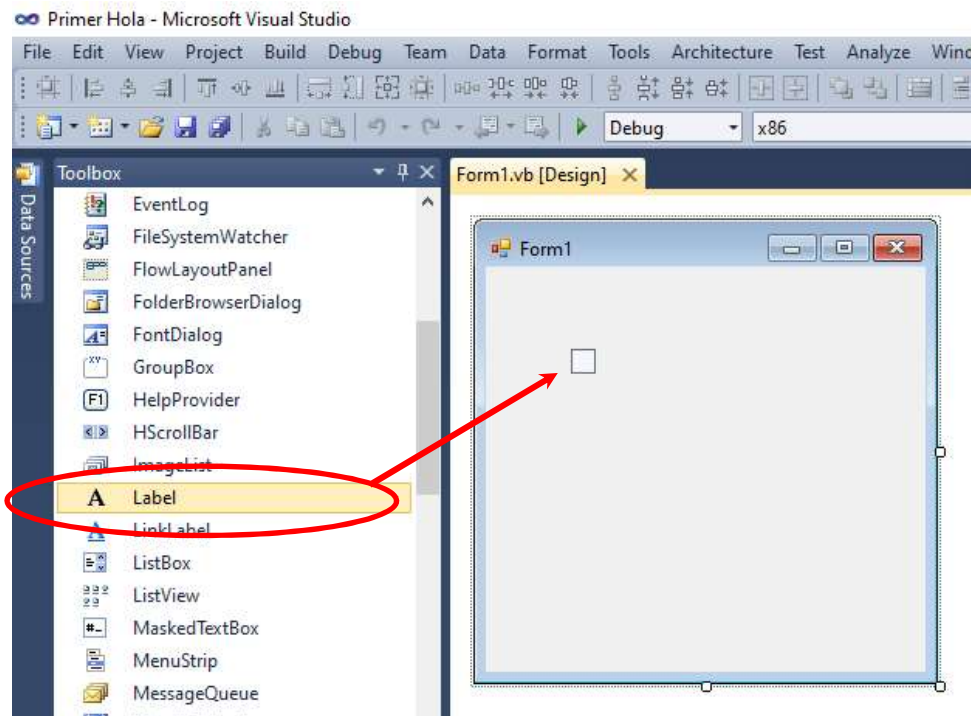


Para facilitar su incursión en el lenguaje C#, es conveniente que el cuadro de herramientas esté siempre visible. Hacer clic en el menú **Ver** y seleccionar **Cuadro de herramientas**.

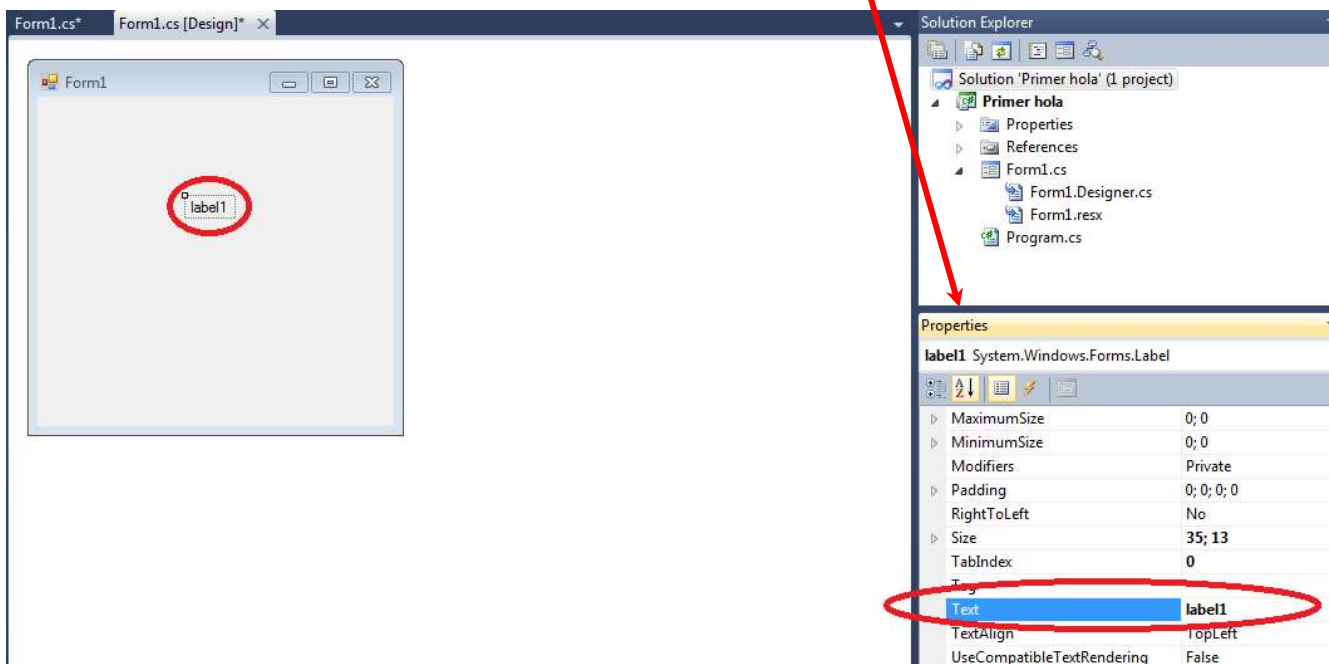
Hacer clic en **Todos los formularios Windows Forms** (dentro de dicho cuadro) para ver la lista de herramientas disponibles.

A continuaciones realizaremos la primera aplicación para mostrar el mensaje de “Hola mundo”

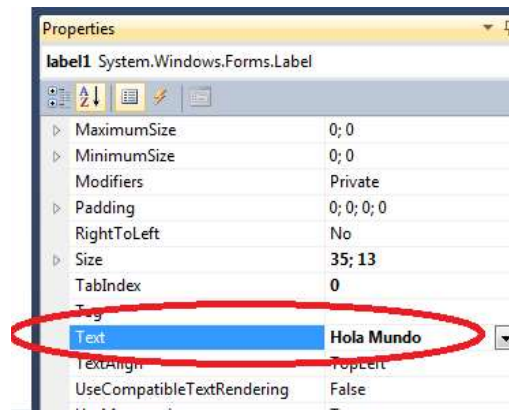
Localizar el cuadro de herramientas, hacer clic en el control **Label** y desplazar el puntero del ratón hacia el formulario. (Hacer clic y, sin soltar el botón, arrastrar hasta el formulario y soltar para crear la etiqueta).



Ahora estableceremos algunas propiedades de la etiqueta: con el botón derecho del ratón sobre la etiqueta hacer clic y seleccionar **Propiedades**. Desplace el ratón hacia la lista **Propiedades** en la parte inferior derecha de la pantalla:



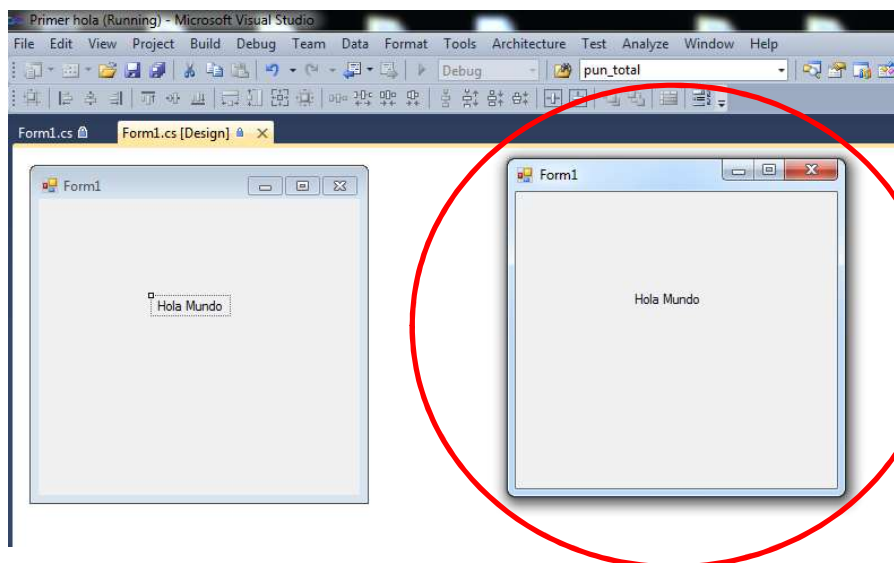
Cambiar la propiedad Text, sustituyendo el texto “label1” por “Hola Mundo”.



Ejecutemos ahora el programa, haciendo clic en la flecha que se encuentra en la parte superior: el botón **Iniciar depuración**:



Aparecerá una nueva ventana. Es el programa que hemos creado. Sólo muestra algo de texto, pero es una verdadera ventana en cuanto a que puede moverla, cambiar su tamaño y cerrarla al hacer clic en el icono **X** que se encuentra en la esquina superior derecha. Experimente con esta ventana, y después ciérrela.



Vaya al menú **Archivo** y seleccione la opción **Guardar todo**.

Puede surgir una ventana nueva de **Guardar Proyecto**, en ese caso asegúrese de que la opción **Crear directorio para la solución** no esté marcada. Deje las demás opciones como están y haga clic en **Guardar**. Cuando vuelva a guardar el proyecto se utilizarán de manera automática las mismas opciones y ya no aparecerá la ventana **Guardar proyecto**.

Ahora sí se puede utilizar el comando **Archivo | Salir** para cerrar el IDE. La próxima vez que utilice C# el nombre del proyecto aparecerá en la Página de inicio, de manera que se podrá abrir con un solo clic, sin necesidad de repetir el trabajo que hicimos para configurar el proyecto.

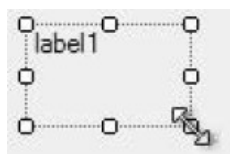
Los controles en tiempo de diseño

En nuestro programa **Primer Hola** colocamos una etiqueta en un formulario y modificamos el texto a mostrar. El propósito principal del ejercicio consistió en repasar los pasos involucrados en la creación de un proyecto. A continuación conoceremos algunos de los fundamentos de los controles y las propiedades.

¿Qué es un control o herramienta? Es un “dispositivo” que aparece en la pantalla, ya sea para mostrar información, para permitir que el usuario interactúe con la aplicación, o ambas cosas. El mismo IDE de C# emplea muchos controles. Por ejemplo, en el ejercicio anterior se utilizó los menús desplegables para guardar proyectos, y el botón **Aceptar** para confirmar acciones. En el caso de las aplicaciones para Windows, el cuadro de herramientas contiene cerca de 70 controles; parte de la tarea de programación implica seleccionar los controles apropiados, situarlos en un formulario y establecer sus propiedades. Esta fase tiene relación con el uso de los controles y se denomina “tiempo de diseño”. Cuando el programa se ejecuta, el usuario interactúa con los controles, es entonces el “tiempo de ejecución”. La labor del programador consiste en crear una **interfaz gráfica de usuario (GUI)** para facilitar dicha interacción.

Veamos ahora cómo se pueden manipular los controles en tiempo de diseño.

Es posible seleccionar un control del cuadro de herramientas y colocarlo en un formulario. La posición inicial que se le asigne no es importante, ya que podemos modificarla con facilidad. También es posible mover el control. Si se coloca el puntero del ratón sobre un control, a su lado aparecerá una flecha con cuatro puntas, como se muestra en la Figura. En ese momento podrá arrastrar el control con el ratón.



Se puede cambiar el tamaño del control. Al hacer clic en un control aparecen varios cuadros pequeños (controladores de tamaño) en sus bordes. Coloque el ratón sobre uno de estos cuadros pequeños y aparecerá una flecha con dos puntas, como se muestra en la Figura.

Arrastre el borde o esquina para modificar el tamaño del rectángulo.

De hecho, el método para cambiar el tamaño depende del control en sí. Por ejemplo, el control tipo etiqueta tiene una propiedad llamada **AutoSize** (determinación automática de tamaño, de acuerdo con las dimensiones del texto introducido) que establecemos como falsa (**False**). Si dejamos la propiedad **AutoSize** como verdadera (**True**), el ancho de la etiqueta se determinará con base al ancho del texto que vaya a mostrar, y la altura se determinará en función del tamaño de la fuente del texto.

Así como los televisores tienen propiedades tales como el color de la carcasa, el tamaño de la pantalla, el canal que está mostrando en un momento dado, su precio y su marca, de igual manera, cada control tiene un conjunto de propiedades que pueden ser ajustadas en tiempo de diseño según los requerimientos. (Luego veremos cómo cambiar una propiedad en tiempo de ejecución)

Después de colocar un control en el formulario, para ver sus propiedades se puede hacer clic con el botón derecho sobre él y seleccionar **Propiedades**; o sólo hacer clic y dirigirse al panel de **Propiedades** a la derecha del IDE. La columna izquierda contiene los nombres de las propiedades, y la columna derecha su valor actual. Para cambiar una propiedad debemos modificar el valor en la columna derecha. En algunas propiedades tal vez sea necesario seleccionar varias opciones adicionales, como al cambiar la configuración de los colores y las fuentes de texto. Algunas veces, cuando el rango de valores a elegir es muy amplio, se requiere abrir una ventana adicional.

Otro de los aspectos vitales de un control es su nombre. Éste no es en sí mismo una propiedad, pero por conveniencia se muestra en la lista de propiedades como (**Name**). Los paréntesis indican que en realidad no es una propiedad.

Cuando colocamos varias etiquetas en un formulario, el IDE selecciona sus nombres de la siguiente manera:

Label1 Label2 Label3...

Estos nombres son aceptables por ahora, pero en los siguientes capítulos veremos que es mejor cambiar los nombres predeterminados de algunos controles por nombres más representativos. Para cambiar el nombre de un control es preciso modificar el texto que está a la derecha de (**Name**) en la lista de propiedades.

1.1. Ejercicio de práctica opcional: Colocar dos Labels en un formulario, moverlos y hacer las siguientes modificaciones en sus propiedades:

Propiedad	Valor
AutoSize	True
Text	Nombre y Edad (respectivamente)
Font	A gusto
BackColor	A gusto

Luego seleccionar el formulario completo y realizar las siguientes tareas, ejecutando el programa después de cada modificación.

- ✓ Cambiar el tamaño del formulario.
- ✓ Alterar su propiedad **Text**.
- ✓ Alterar su propiedad **BackColor**.

Después de realizar cada modificación ejecutar el programa, observar el efecto y detener la ejecución haciendo clic en el botón **X** en la esquina superior derecha de la ventana.

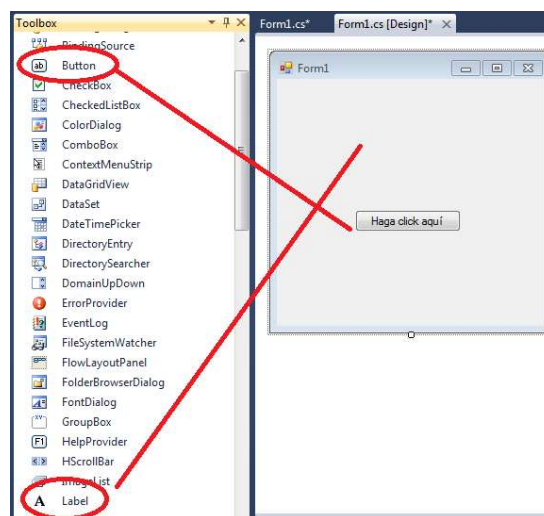
Los eventos y control button

Ya que el programa que creamos en el ejercicio anterior no es muy representativo, dado que siempre muestra las mismas palabras y el usuario no puede interactuar con él, enriqueceremos este programa de manera que al hacer clic en un botón, aparezca un mensaje de texto. Éste es un ejemplo de cómo usar un evento.

Casi todos los eventos son puestos en acción por el usuario, y se generan cuando éste manipula un control de alguna forma en tiempo de ejecución. Cada control tiene varios eventos a los que puede responder, como el clic de un botón del ratón, un doble clic o la colocación del puntero del ratón sobre el control. Otros tipos de eventos no tienen su origen en el usuario; por ejemplo, la notificación de que una página Web ha terminado de descargarse.

He aquí la forma en que crearemos la interfaz de usuario:

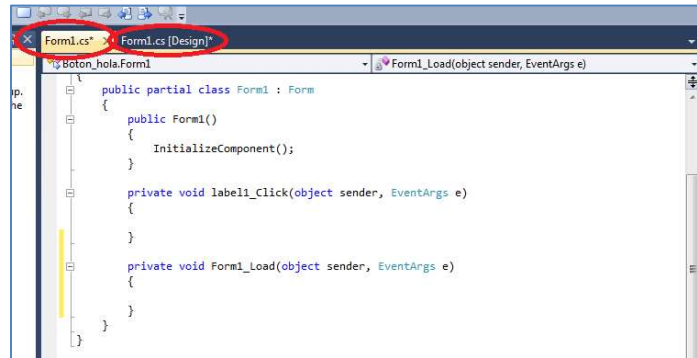
- ✓ Crear un nuevo proyecto llamado **Botón hola** y colocar un Label y un Button en el formulario. La posición en que lo haga no es importante.
- ✓ Escriba **Haga clic aquí** en la propiedad **Text** del botón.
- ✓ Modifique la propiedad **Text** de la etiqueta, de forma que aparezca sin contenido (borrar el texto "label1").



Ejecutar el programa, aunque todavía no está completo. Observar que se puede hacer clic en el botón, y que éste cambia su aspecto ligeramente para confirmar que está siendo oprimido; no ocurre nada más. Cerrar el formulario.

Veamos ahora cómo detectar el evento del clic. Haga doble clic en el botón dentro del formulario de diseño. A continuación se abrirá un nuevo panel de información, como el siguiente: Observar las fichas de la parte superior:

Form1.cs
Form1.cs [Diseño]



Se puede hacer clic en esas fichas para cambiar de panel. El panel Form1.cs muestra un programa de C#. A esto se le conoce como “texto del programa”, o “código” de C#. Modifiquemos este código utilizando el editor del IDE.

Desplácese por el código hasta que encuentre la siguiente sección:

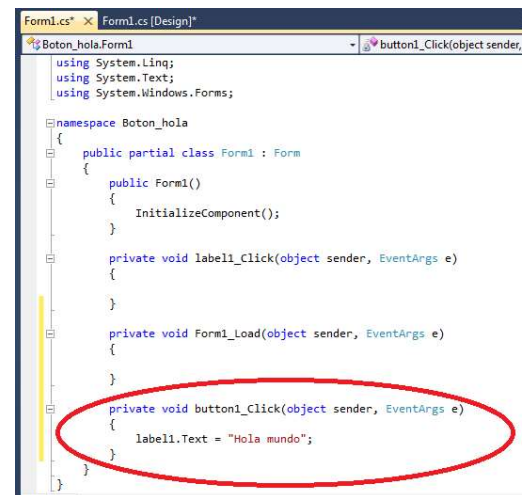
```
private void button1_Click(object sender, EventArgs e)
{
}
```

A esta sección de código se le conoce como *método*. El nombre del método es **button1_Click**.

Cuando el usuario haga clic sobre el botón **button1** en tiempo de ejecución, se llevará a cabo, o “ejecutará”, cualquier instrucción que coloquemos entre las dos líneas anteriores.

En este programa específico utilizaremos una instrucción para colocar el mensaje **Hola mundo** en el texto de la etiqueta **label1**. La instrucción para realizar esta acción es:

```
label1.Text = "Hola Mundo";
```



Escriba la instrucción exactamente como se muestra aquí; hágalo entre las líneas { y }. En los siguientes capítulos explicaremos el significado exacto de líneas como la anterior (que implican una “instrucción de asignación”).

El siguiente paso es ejecutar el programa. Para ello hay dos posibilidades:

Si el programa se ejecuta correctamente, al hacer clic en el botón observará que aparece el mensaje **Hola mundo** en la etiqueta.



La otra posibilidad es que el programa no se ejecute debido a un error. En ese caso, se podrá tildar la opción de no volver a mostrar el mensaje de error, y se subrayarán las líneas de código a revisar, pero se deberá solucionar en lo posterior para una adecuada ejecución.

El cuadro de mensaje

Anteriormente utilizamos el control tipo etiqueta para mostrar texto en la pantalla, pero también podemos usar un cuadro de mensajes para lograrlo. Este control no aparece en el cuadro de herramientas, debido a que no ocupa un espacio permanente en un formulario; sólo aparece cuando es requerido. El siguiente es un fragmento de código que muestra el mensaje **Hola mundo** dentro de un cuadro de mensajes al hacer clic en un botón:

```
private void button1_Click(object sender, EventArgs e)
{
    MessageBox.Show("Hola mundo");
}
```



Al ejecutar el programa y hacer clic en el botón se despliega el cuadro/ventana de mensajes, y debemos hacer clic en el botón **Aceptar** para que desaparezca. Esta característica implica que los cuadros de mensajes se utilizan para mostrar mensajes importantes que el usuario no puede ignorar.

Para utilizar un cuadro de mensajes escriba una línea como la anterior, pero utilice su propio mensaje dentro de las comillas dobles. Más adelante explicaremos el propósito de la instrucción **Show** con más detalle.

1.2. Ejercicio de práctica opcional: Crear un nuevo proyecto llamado **Demo** y colocar tres botones en el formulario. Asignar los dígitos **1**, **2** y **3**, respectivamente, a sus propiedades de texto. Crear tres etiquetas y establecer sus propiedades de texto como **A**, **B** y **C**, respectivamente. Luego colocar el código necesario en los métodos apropiados de los botones, de manera que:

- (a) al hacer clic en **button1** el texto de todas las etiquetas cambie a **Sí**;
- (b) al hacer clic en **button2** el texto de todas las etiquetas cambie a **No**;
- (c) al hacer clic en **button3** los valores de texto regresen a **A**, **B**, **C**.

1.3. Ejercicio de práctica opcional: Agregar un botón al formulario del proyecto anterior (**Demo**), de modo que al presionarlo aparezca un cuadro de mensaje con el texto "Adiós Mundo".

1.4. Ejercicio de práctica opcional: En este ejercicio se debe utilizar la propiedad **Visible** de un control, la cual puede establecerse como **true** o **false** (no utilice mayúsculas). Por ejemplo, el siguiente código hace invisible la etiqueta **label1**:

```
label1.Visible = false;
```

Agregar, entonces, al programa **Demo** dos botones. Al hacer clic en un botón las etiquetas deberán hacerse invisibles, y al hacer clic en el otro deberán hacerse visibles nuevamente.