		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

## TEMA N° 2. COMENZAR A PROGRAMAR CON C#


El programa que se creará a continuación desplegará el mensaje “Hola Mundo” en la pantalla de su computadora. Independientemente de lo anterior, deberá seguir estos pasos generales para crear cualquier programa.

### ENTORNO DE DESARROLLO DE C#



Figura 2.2 La ventana Nuevo proyecto.

- Abra el IDE. A continuación aparecerá la Página de inicio
- Haga clic en el vínculo Crear: Proyecto... En la Figura 2.2 se muestra la ventana Nuevo proyecto que aparecerá en respuesta.
- Asegúrese de que esté seleccionada la plantilla Aplicación de Windows Forms. Elija un nombre para su proyecto, mismo que se convertirá también en la identificación de una carpeta. Le recomendamos utilizar sólo letras, dígitos y espacios. En nuestro caso utilizamos el nombre Primer Hola.
- Haga clic en Aceptar. A continuación aparecerá un área de diseño similar (aunque no necesariamente idéntica) a la que se muestra en la Figura 2.4.
- Para facilitar su incursión en el lenguaje C#, es conveniente que el cuadro de herramientas esté siempre visible. Haga clic en el menú Ver y seleccione Cuadro de herramientas. Ahora haga clic en el símbolo de “chincheta” del cuadro de herramientas, como se muestra en la Figura 2.3; el cuadro de herramientas quedará fijo y abierto de manera permanente. Haga clic en Todos los formularios Windows Forms para ver la lista de herramientas disponibles. Ahora su pantalla se verá casi idéntica a la de la Figura 2.4.

		Evaluación del módulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

Como se mencionó previamente, los pasos anteriores son comunes en la realización de cualquier proyecto. Realice ahora una tarea específica para este proyecto en particular.

Observe el formulario y el cuadro de herramientas ilustrados en la Figura 2.4. A continuación seleccionará un control del cuadro de herramientas, y lo colocará en el formulario. Éstos son los pasos a seguir:

- Localice el cuadro de herramientas y haga clic en el control Label.
- Desplace el puntero del ratón hacia el formulario. Haga clic y, sin soltar el botón, arrastre para crear una etiqueta como en la Figura 2.5.
- Ahora establecerá algunas propiedades de la etiqueta: haga clic con el botón derecho del ratón sobre la etiqueta y seleccione Propiedades. Desplace el ratón hacia la lista Propiedades en la parte inferior derecha de la pantalla, como se muestra en la Figura 2.6.

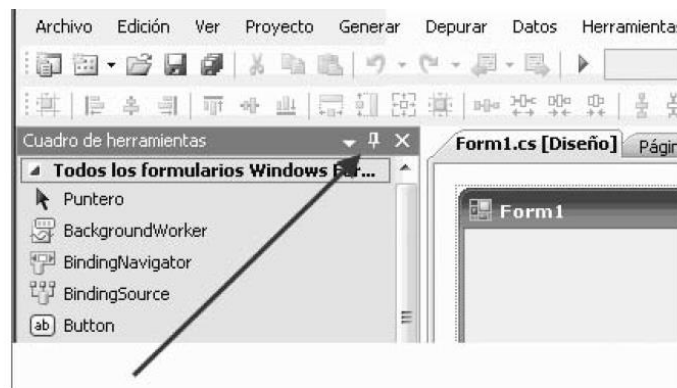


Figura 2.3 El cuadro de herramientas queda fijo y abierto.

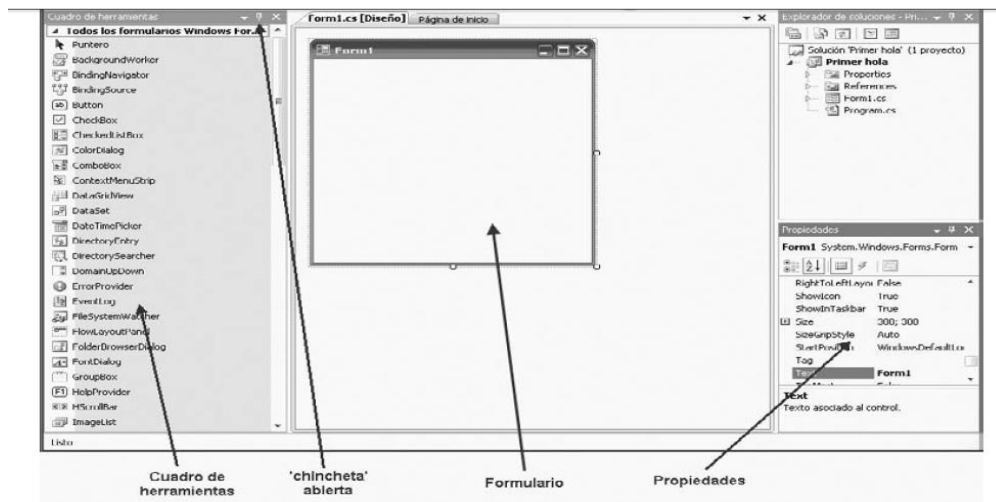


Figura 2.4 El IDE en tiempo de diseño.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

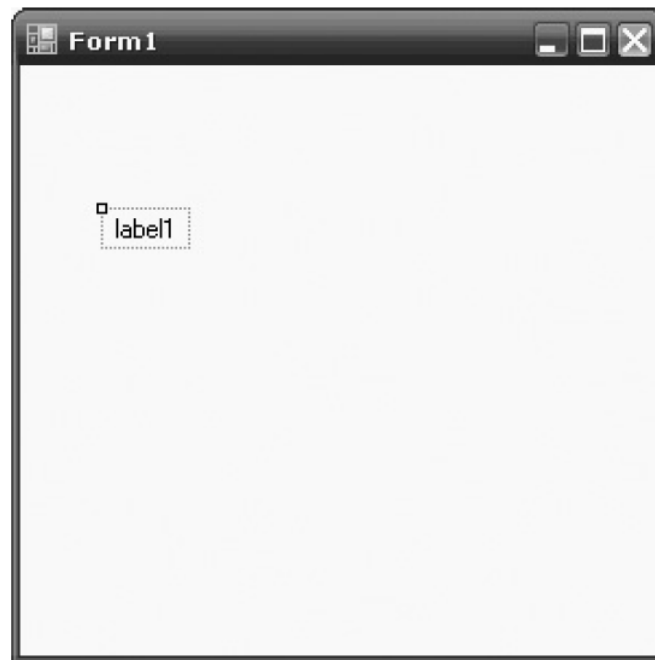


Figura 2.5 Se agregó una etiqueta al formulario.

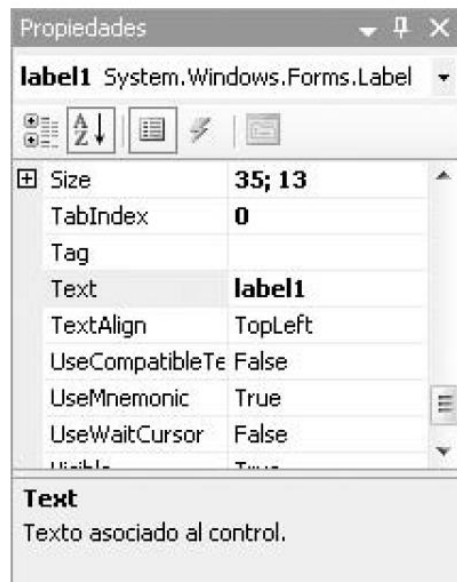



Figura 2.6 Propiedades de la etiqueta.




Figura 2.7 Haga clic en la flecha para ejecutar el programa.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

- Desplácese hasta la propiedad Text y sustituya el texto label1 por Hola Mundo.
- Ejecute ahora el programa, haciendo clic en la flecha que se encuentra en la parte superior del IDE (Figura 2.7); se trata del botón Iniciar depuración.
- Aparecerá una nueva ventana, como se muestra en la Figura 2.8. Es el programa que usted ha 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.
- Para guardar su programa y poder usarlo más adelante:
- Vaya al menú Archivo y seleccione la opción Guardar todo (en adelante, para indicar este tipo de acciones utilizaremos una forma abreviada como ésta: Archivo | Guardar todo).
- A continuación aparecerá la ventana Guardar proyecto, como se muestra en la Figura 2.9. 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 puede utilizar el comando Archivo | Salir para cerrar el IDE. La próxima vez que utilice C# el nombre de su proyecto aparecerá en la Página de inicio, de manera que podrá abrirlo con un solo clic, sin necesidad de repetir el trabajo que hizo para configurar el proyecto.



Figura 2.8 El programa Primer hola en ejecución.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

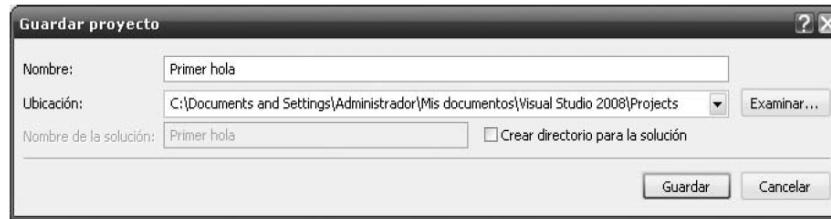


Figura 2.9 La ventana Guardar proyecto, con la opción Crear directorio desactivada.

En el programa **Primer Hola** se colocó una etiqueta en un formulario y se modificó el texto a desplegar. El propósito principal del ejercicio consistió en repasar los pasos involucrados en la creación de un proyecto.


A continuación, se conocerán algunos de los fundamentos de los controles y las propiedades.

**¿QUÉ ES UN CONTROL?** 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 usted 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. A diferencia del “tiempo de ejecución”, 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. 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 se puede modificar con facilidad.
- También es posible mover el control. Si coloca el puntero del ratón sobre un control, a su lado aparecerá una flecha con cuatro puntas, como se muestra en la Figura 2.10. 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 2.11.

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

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

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**), como se ilustra en la Figura 2.11.

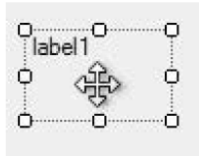


Figura 2.10 Los controles pueden moverse ...

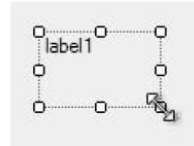


Figura 2.11 ... y redimensionarse según convenga.


Si se deja 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.

Otros controles permiten variar el ancho arrastrando los controladores con el ratón, pero no la altura (que se establece de acuerdo con las fuentes). Algunos controles como los botones permiten cambiar su tamaño en cualquier dirección. En general el tamaño de los controles depende del uso que se les vaya a dar, por lo que le recomendamos experimentar con ellos. Sin embargo, y dado que las etiquetas son tan comunes, no debemos olvidar su propiedad **AutoSize**.

A continuación se examinarán las propiedades. He aquí una analogía: 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 para satisfacer nuestros requerimientos. Más adelante se verá cómo cambiar una propiedad en tiempo de ejecución.

Después de colocar un control en el formulario, para ver sus propiedades hay que hacer clic con el botón derecho sobre él y seleccionar **Propiedades**; de esta manera se desplegará la ventana de propiedades del control elegido. 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.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

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:

**etiqueta1 etiqueta2 etiqueta3 ...**

Estos nombres son aceptables por ahora, pero más adelante se verá 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.

### LOS EVENTOS Y EL CONTROL BUTTON

El programa que se creó en el ejercicio anterior no es muy representativo, dado que siempre muestra las mismas palabras y el usuario no puede interactuar con él. Enseguida se enriquecerá este programa de manera que aparezca un mensaje de texto cuando el usuario haga clic en un botón.


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.

En el programa que se creará a continuación se detectará un evento (el clic de un botón); después se hará que se despliegue un mensaje de texto en una etiqueta.

He aquí la forma en que se creará la interfaz de usuario:

- Cree un nuevo proyecto llamado Botón hola.
- Coloque una etiqueta y un botón 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.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

- Ejecute el programa, aunque todavía no está completo. Observe que puede hacer clic en el botón,
- y que éste cambia su aspecto ligeramente para confirmar que está siendo oprimido; no ocurre nada más. Cierre 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 que se muestra en la Figura 2.12. Observe las fichas de la parte superior:

Form1.cs    Página de inicio    Form1.cs [Diseño]

Usted 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.

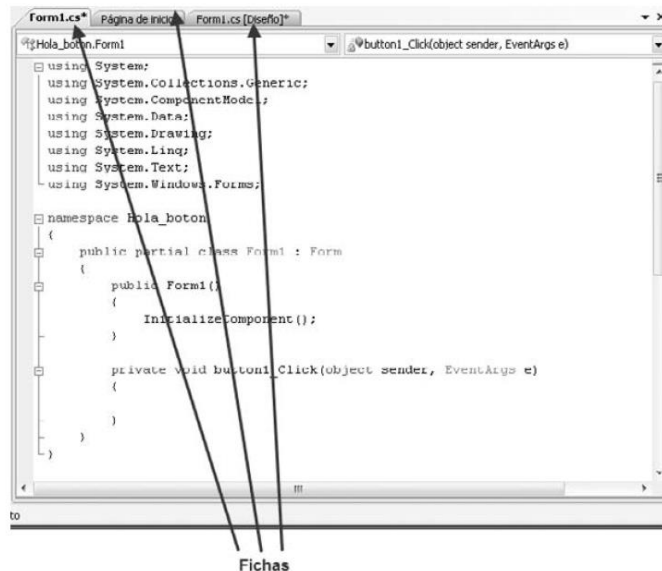



Figura 2.12 Código de C# en el panel de edición.

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.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

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 }.

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.




Figura 2.13 Acción errónea. Marque la casilla de verificación y haga clic en No.

La otra posibilidad es que el programa no se ejecute debido a un error. En este caso C# mostrará un mensaje como el de la Figura 2.13. Marque la casilla de verificación de la opción **No volver a mostrar este cuadro de diálogo**, y haga clic en **No** para continuar. El mensaje no volverá a aparecer; en lo sucesivo la única evidencia del error será un subrayado en el texto del código.

De todas formas hay que corregir el error: revise el código, corrija cualesquiera errores de escritura, y ejecute el programa de nuevo. Más adelante hablaremos sobre los errores con más detalle.

He aquí las nuevas características de este programa:

- Puede responder al clic en un botón. Al proceso de colocar código de manera que se lleve a cabo la acción correspondiente cuando ocurra un evento se le conoce como “manejar” el evento.
- Modifica el valor de la propiedad de un control en tiempo de ejecución. Anteriormente sólo hacíamos esto en tiempo de diseño. Ésta es una parte muy importante de la programación, ya que a menudo tenemos que mostrar resultados colocándolos en cierta propiedad de un control.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

### DOCUMENTACIÓN DE LOS VALORES DE LAS PROPIEDADES

En este libro necesitamos proveerle los valores de las propiedades que emplearemos en los controles.

Cuando las propiedades sean pocas podremos explicarlas en unos cuantos enunciados, pero si la cantidad es mayor utilizaremos una tabla. En general tenemos varios controles; cada control tiene varias propiedades y cada propiedad tiene un valor. Tenga en cuenta que hay una gran cantidad de propiedades para cada control, pero sólo listaremos aquellas que usted tenga que modificar. Las demás mantendrán su valor predeterminado. He aquí un ejemplo:

Control	Propiedad	Valor
label1	Text	Hola mundo
label1	BackColor	Red
button1	Text	Haga clic aquí

Los valores listados corresponden a los siguientes controles:

- **label1** cuyo texto es **Hola mundo**, y cuyo color de fondo es rojo;
- **button1** cuyo texto es **Haga clic aquí**.

Tenga cuidado al escribir los nombres de las propiedades. Debe hacerlo con letra mayúscula inicial, y sin espacios.

### • ERRORES EN LOS PROGRAMAS


Es común que ocurran errores en los programas; por fortuna, el IDE puede detectar algunos por nosotros. He aquí un ejemplo de error:

**label1.Txt = "Hola mundo"**

Si usted escribe esta línea y ejecuta el programa, el IDE la subrayará. Al colocar el puntero del ratón sobre la parte subrayada aparecerá una explicación del error. Tenga en cuenta que la explicación podría ser difícil de comprender (incluso para un experto), o que tal vez el error se encuentre en otra parte del programa.

Sin embargo, el primer paso deberá consistir siempre en inspeccionar el área subrayada para ver si hay errores de escritura. A medida que vaya aprendiendo más sobre el lenguaje C# mejorará su habilidad para detectar errores.

Una vez corregidos los errores subrayados hay que ejecutar el programa. De hecho, antes de que esto ocurra se activa otro programa llamado compilador, el cual realiza comprobaciones en el código para detectar, quizá, más errores. Sólo

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

hasta que se hayan corregido todos los errores de compilación su programa podrá ejecutarse.

Los errores de compilación se muestran en la ventana **Lista de errores**, la cual se abre debajo del código. Al hacer doble clic en el error dentro de la ventana de resultados, el IDE nos llevará al punto del código donde se debe corregir el error.

Cualquiera que empiece a escribir programas enfrentará muchos errores de compilación. No se desanime; esto es parte del proceso de aprendizaje. Tras corregir los errores de compilación el programa podrá ejecutarse.

En este punto podríamos notar que lo hace de manera incorrecta, lo cual querría decir que hay un “error en tiempo de ejecución”, o *bug*. Dichos errores son más difíciles de corregir, y en consecuencia es necesario realizar una depuración.

### **FUNCIONES DEL EDITOR**

El editor no se limita a mostrar lo que escribe el programador; las siguientes son algunas de sus virtudes adicionales:


- Ofrece las operaciones estándar de cortar, copiar y pegar en el portapapeles, gracias a lo cual usted podrá copiar código de otros programas de Windows.
- Para escribir programas más grandes se requieren controles y eventos adicionales. Las listas desplegables que están en la parte superior del editor nos permiten seleccionar controles, además de un evento específico para dicho control. El editor se coloca de manera automática en el método apropiado.
- El editor desplegará el código de C# que se va tecleando de acuerdo con ciertas reglas. Por ejemplo, se insertarán espacios de manera automática alrededor del símbolo =.
- Algunas líneas necesitarán sangrías —desplazamiento del texto a la derecha—, para lo cual hay que insertar espacios adicionales en blanco. Por lo general esto se hace en intervalos de cuatro espacios. Como veremos en capítulos posteriores, el código de C# consiste de secciones dentro de otras secciones, y las sangrías nos ayudan a indicar en dónde inicia y termina cada sección.

Al proceso de aplicar sangrías a un programa también se le conoce como *darle formato*. El IDE puede hacerlo de manera automática a través del menú:

**Edición | Avanzado | Dar formato al documento** Es conveniente dar formato al código con frecuencia.

- Cada tipo de control cuenta con un conjunto de propiedades particular. Si usted escribe el nombre de un control seguido de un punto, como se muestra a continuación:

**label1.**

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

y espera un poco, el editor le proporcionará una lista de las propiedades de la etiqueta label1. Si esta herramienta no se encuentra configurada en su sistema, puede activarla seleccionando la opción:

### Herramientas | Opciones | Editor de texto | C#

y marcando después la casilla de verificación **Lista de miembros automática**.

- Las secciones del código en las que no estemos trabajando pueden contraerse en una sola línea de código. Para ello hay que hacer clic en el pequeño símbolo “-” que está a la izquierda del editor. Para expandir una sección haga clic en el símbolo “+”. Encontrará estos símbolos al lado de líneas como la siguiente:

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

No es preciso que usted recuerde todas estas herramientas, pues siempre están al alcance de su mano para ayudarlo.

Una de las tareas relacionadas con la distribución del código que no se realizan de manera automática es la división de líneas extensas. Para asegurar que toda una línea esté visible en la pantalla podemos elegir un lugar adecuado para dividirla y oprimir la tecla **Enter**. El IDE aplicará una sangría de cuatro espacios a la segunda parte de la línea. He aquí un ejemplo:

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


Si presionamos la tecla **Enter** después de la coma, el código aparecerá así:

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

Muchas veces el IDE crea líneas muy extensas que, por supuesto, no deben contener errores, de manera que es mejor no dividirlas. Sin embargo, a medida que sea más experimentado en su manejo de C#, algunas de estas largas líneas serán de su propia creación, y resultará muy conveniente dividirlas de manera que pueda verse todo el texto a la vez. Con esto también se mejora la legibilidad del código impreso (conocido como listado).

### EL CUADRO DE MENSAJES

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.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

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");
}
```

La Figura muestra lo que ocurre al ejecutar el programa y hacer clic en el botón: se despliega el cuadro de mensajes, y debemos hacer clic en el botón Aceptar para que desaparezca.




### **AYUDA**

El sistema de ayuda funciona con base en dos posibles fuentes: una local, ubicada en su propia computadora, y la otra a través de Internet. La fuente de Internet se actualiza con frecuencia; sin embargo, la versión local es suficiente cuando se empieza a trabajar con C#: En caso de que usted no cuente con una conexión a Internet, el programa utilizará automáticamente la ayuda local.

Si desea especificar de dónde debe provenir la ayuda, puede reconfigurar el sistema de la siguiente forma:

1. En la ventana principal de C#, seleccione **Ayuda | Buscar...**
2. En la nueva ventana de ayuda seleccione **Ayuda | Ayuda sobre la Ayuda**. A continuación aparecerán varias opciones sobre cuál fuente de ayuda tendrá mayor prioridad.

El sistema de ayuda de C# es extenso, pero si usted es nuevo en materia de programación tal vez la información que contiene le resulte complicada. Es más provechoso cuando se tiene algo de experiencia con C# y se requiere un detalle técnico preciso.

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

Las opciones más útiles del menú Ayuda son **Índice** y **Buscar**, ya que nos permiten escribir cierto texto para que el sistema localice las páginas que puedan servirnos. La diferencia es que **Índice** sólo busca en los títulos de las páginas, mientras que **Buscar** también examina el contenido de las mismas.

### FUNDAMENTOS DE PROGRAMACIÓN

- Los controles pueden colocarse en un formulario en tiempo de diseño.
- Es posible establecer las propiedades de los controles en tiempo de diseño.
- Los programas son capaces de modificar las propiedades en tiempo de ejecución.
- Cuando ocurre un evento (como hacer clic en un botón), el sistema de C# utiliza el método
- apropiado, pero es el programador quien debe colocar código dentro del método para manejar ese evento.

### ERRORES COMUNES DE PROGRAMACIÓN

- Olvidar dar por terminada la ejecución de su programa antes de tratar de modificar el formulario o el código.
- Confundir el formulario en tiempo de diseño con el formulario en tiempo de ejecución.

### SECRETOS DE CODIFICACIÓN

En el código de C#, para referirnos a la propiedad de un control utilizamos su nombre seguido por un punto y por el nombre de la propiedad, como en:

**label1.Text**

- A una sección de código entre:

```
private void botonNumero_Click(Object sender, EventArgs e)
{
}
```

se le conoce como método.


Los cuadros de mensaje no se colocan directamente en los formularios. Para hacer que aparezcan en pantalla debemos usar la siguiente instrucción:

**MessageBox.Show("Aquí va el texto que usted desee");**

### NUEVOS ELEMENTOS DEL LENGUAJE

Una introducción a las propiedades, métodos y eventos.

### NUEVAS CARACTERÍSTICAS DEL IDE

		Evaluación del modulo		
CÓDIGO SI316	ASIGNATURA: PROGRAMACION II	PRELACIÓN	UC	<b>Profesor: Guido R. Portillo</b>

- Los programas están contenidos en proyectos.
- El IDE crea una carpeta que contiene los archivos necesarios para un proyecto.
- Es posible mover y cambiar el tamaño de los controles en los formularios.
- El cuadro de herramientas contiene una amplia diversidad de controles.
- Al hacer clic con el botón derecho del ratón sobre un control podemos seleccionar sus propiedades.
- Al hacer doble clic en un botón en tiempo de diseño se crean métodos para manejar eventos.

### **RESUMEN**

Parte de la tarea de programación implica colocar controles en formularios y establecer sus propiedades iniciales. El IDE de C# realiza esta tarea directamente, pero es necesario que practique con el IDE y lea sobre él.