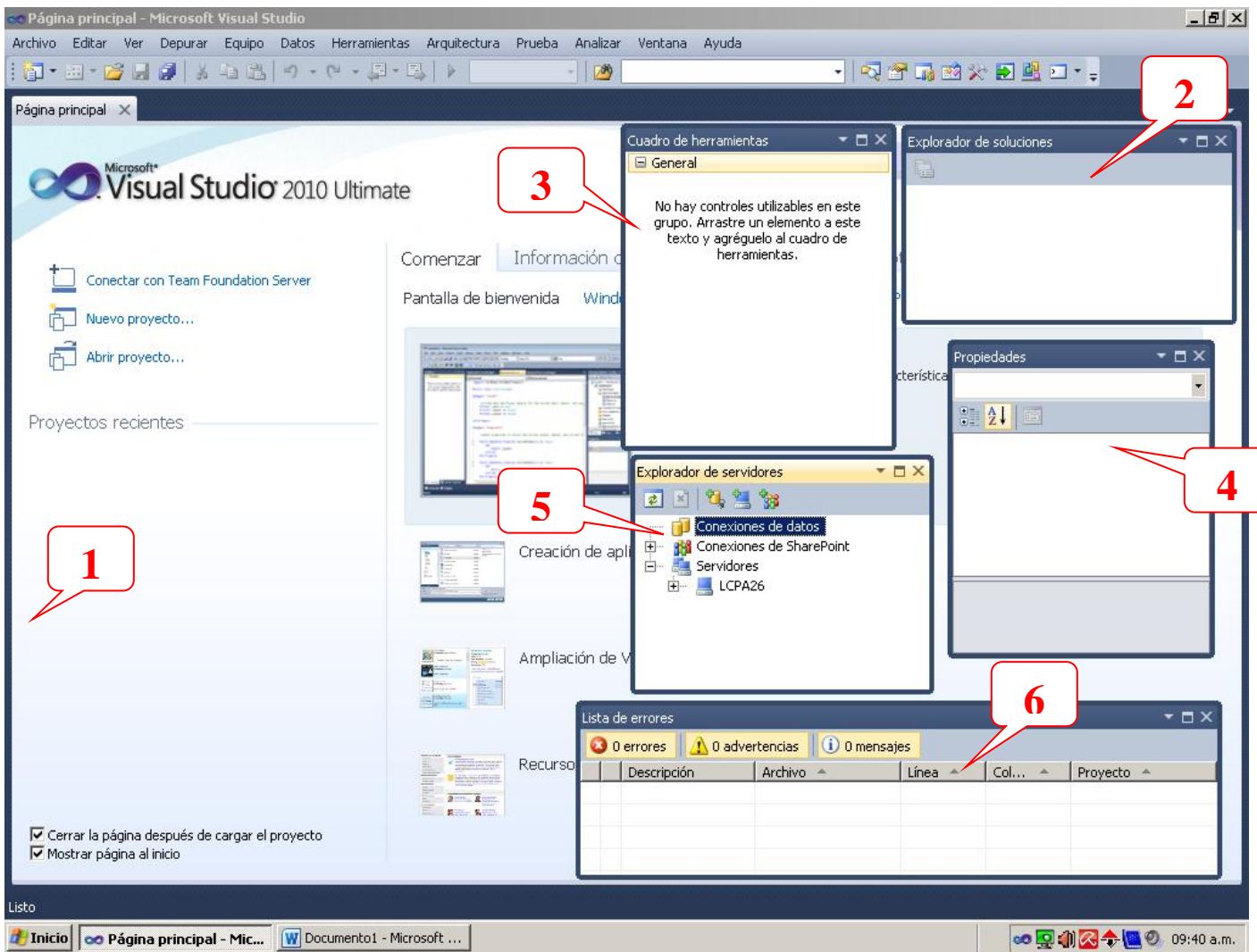


MANUAL:
VISUAL STUDIO®

Visual Estudio 2010



1.-Ventana de inicio: entorno de desarrollo integrado (IDE) que se compone de varios elementos: la barra de menús, la barra de herramientas Estándar, varias ventanas de herramientas que se acoplan u ocultan automáticamente a la izquierda, en la parte inferior y a la derecha, así como en el espacio del editor.

2.-Explorador de Soluciones: muestra soluciones, sus proyectos y los elementos incluidos en dichos proyectos. En el Explorador de soluciones, puede abrir archivos para editar, agregar nuevos archivos a un proyecto y ver las propiedades de las soluciones, proyectos y elementos.

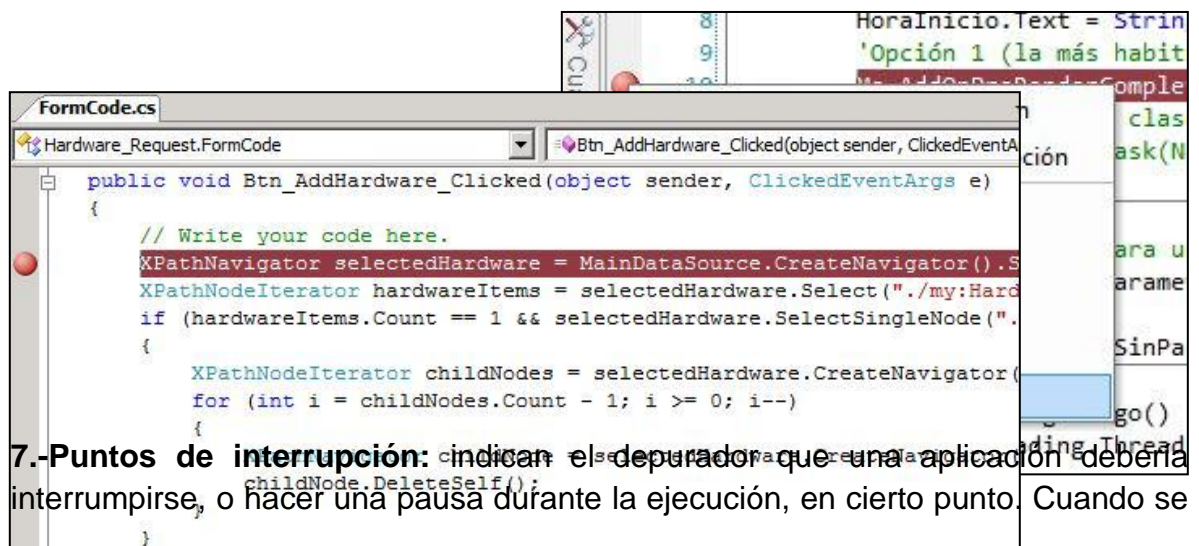
3.-Cuadro de Herramientas: muestra iconos para los controles y otros elementos que se pueden agregar a los proyectos Visual Studio. Se puede arrastrar a una

vista de diseño, o bien pegar en un editor de código en el entorno de desarrollo integrado. Cualquier acción agrega el código básico para crear una instancia del elemento en el archivo del proyecto activo y solo muestra los elementos correspondientes al tipo de archivo en el que se está trabajando.

4.- Propiedades: Utilice esta ventana para ver y cambiar las propiedades y eventos en tiempo de diseño de los objetos seleccionados que están ubicados en editores y diseñadores también se puede utilizar la ventana Propiedades para editar y ver las propiedades de los archivos, proyectos y soluciones también muestra distintos tipos de campos de edición, según las necesidades de una propiedad determinada estos campos de edición incluyen cuadros de edición, listas desplegables y vínculos a cuadros de diálogo de editor personalizado.

5.- Explorador de servidores: es la consola de administración de servidores de Visual Studio esta ventana sirve para abrir conexiones a datos y conectar con servidores para explorar sus bases de datos y servicios de sistema.

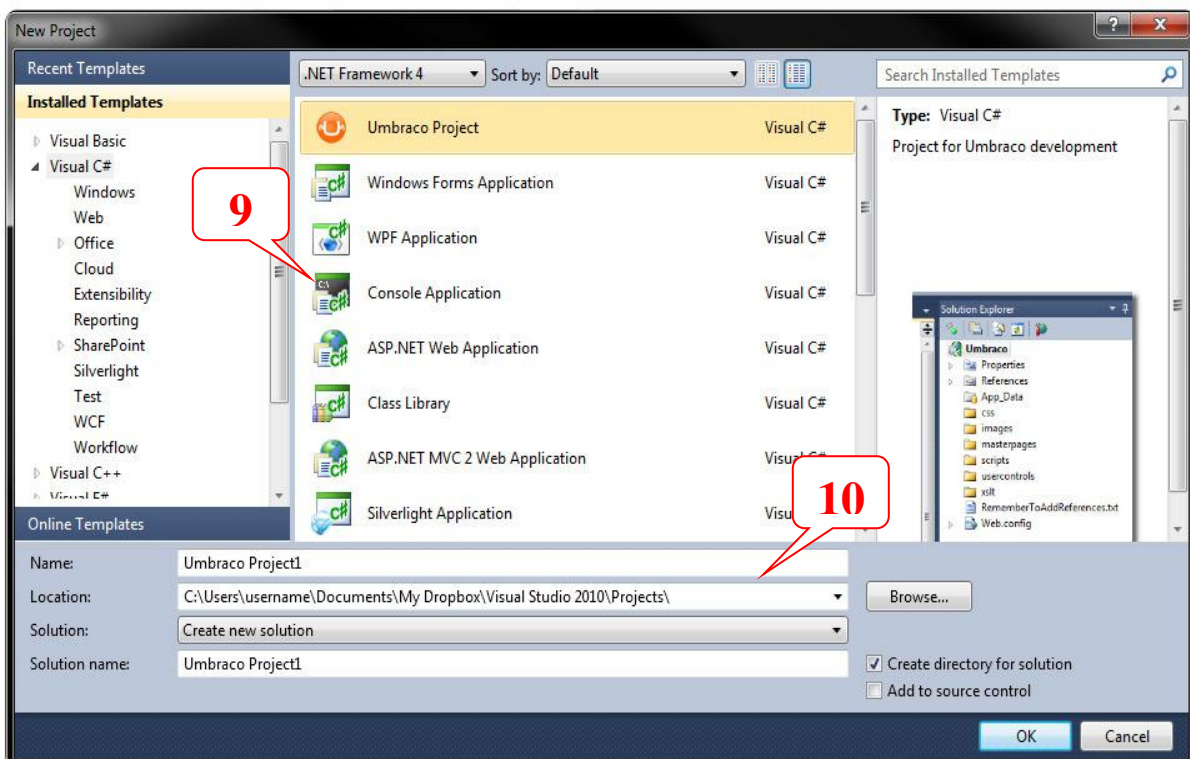
6.- Lista de Errores: muestra los errores, advertencias y mensajes que se generan cuando edita y compila código, busca errores de sintaxis detectados por IntelliSense, errores de implementación, ciertos errores de análisis estático y errores detectados mientras se aplican directivas, sirve para solucionar cualquier mensaje de error para abrir el archivo donde se produce el problema y desplazarse a la ubicación del error.



7.- Puntos de interrupción: indican al depurador que una aplicación debería interrumpirse, o hacer una pausa durante la ejecución, en cierto punto. Cuando se

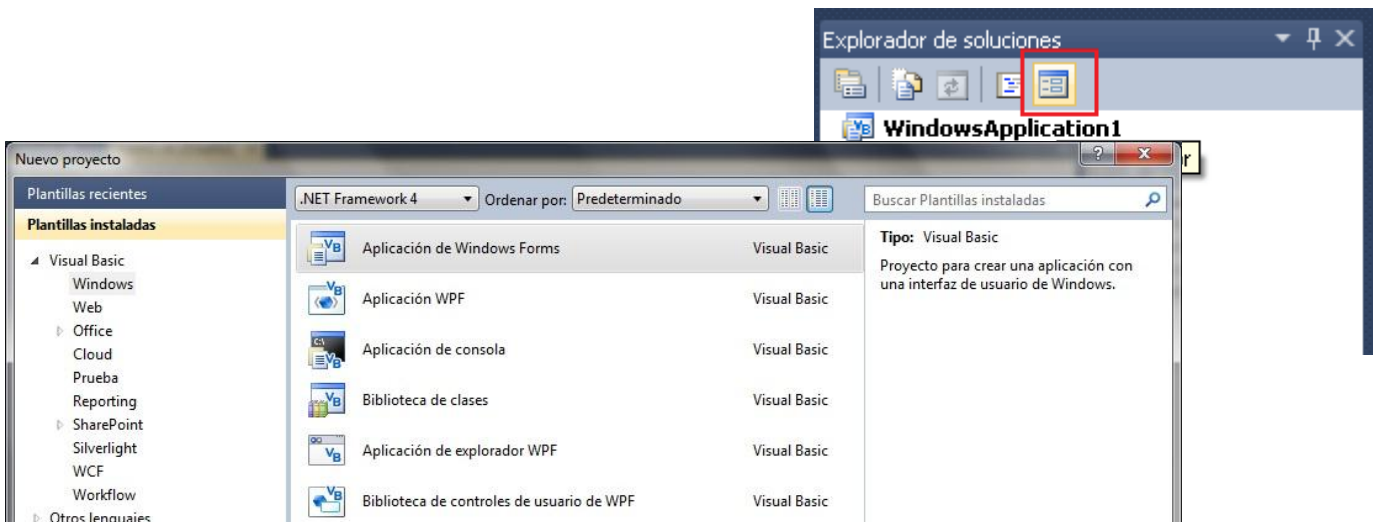
produce una interrupción, se dice que el programa y el depurador están en modo de interrupción. Para obtener más información, vea Interrumpir la ejecución.

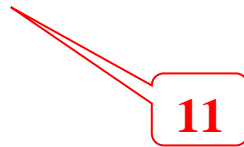
8.- Puntos de seguimiento: son una nueva característica del depurador de Visual Studio. Un punto de seguimiento es un punto de interrupción con una acción personalizada asociada. Cuando se alcanza un punto de seguimiento, el depurador realiza la acción asociada al punto de seguimiento en lugar de, o además de, interrumpir la ejecución del programa.



9.- Creación de un Proyecto: Simplifica el flujo de trabajo desde dentro de Visual

10.-Especificaciones del proyecto: Studio. En esta sección agregamos elementos básicos de nuestro proyecto como nombre y la ubicación donde estará.

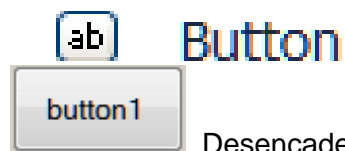




11.-Formularios: Se llama formulario a una plantilla o página con espacios vacíos que han de ser rellenados con alguna finalidad, en visual es una clase o instancia.

CONTROLES, PROPIEDADES Y METODOS

1.- Button



Desencadena un evento cuando un usuario hace clic sobre el.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
--------	-------------	--------	-------------

Name	Indica el nombre utilizado en el código para identificar el objeto.
AccessibleDescription	Descripción que proporciona a clientes con discapacidades para facilitar el acceso.
AccessibleName	Nombre que se proporciona a clientes con discapacidades para facilitar al acceso.
AccessibleRole	Función que se proporciona a clientes con discapacidades para facilitar el acceso.
AllowDrop	Indica si el control puede aceptar datos que el usuario arrastre.
Anchor	Define a que bordes del contenedor está enlazado
AutoElipsis	Habilitar el tratamiento automatico del texto que se estiende mas alla del ancho del botón.
Autosize	Especifica si un control cambiara automáticamente de tamaño para ajustarse a su contenido.
AutoSizeMode	Espacifica el modo mediante el que el elemento de interfaz del usuario cambia de tamaño automáticamente.
BlackColor	El color de fondo del componente.
BackgroundImage	Imagen de fondo utilizada para el control.
BackgroundImageLayout	Diseño de imagen de fondo utilizada para el componente.
CausesValidation	Indica si este componente genera eventos de validación.
ContextMenuStrip	Menu contextual que se muestra cuando el usuario hace clic con el botón secundario en el control.
Cursor	Cursor que aparece al pasar el puntero por el control.
DialogResult	El resultado producido por el cuadro de dialogo hacer clic en el botón de un formulario modal.
Dock	Define los bordes de control enlazados al contenedor.
Enable	Indica si el control esta habilitado.
FlatApperance	Para los botones cuyo FlatStyle es FlatStyle.Flat, determina la apariencia del borde y los colores utilizados para indicar el estado da activación el estado del mouse.
FlatStyle	Determinala apariencia del control cuando un usuario mueve el mmouse sobre el control y hace clic.
Font	Fuente utilizada para mostrar texto en el control.
ForeColor	Obtiene o establece el color de primer plano del control.
Location	Obtiene o establece las coordenadas de la esquina superior izquierda del control en relación con la esquina superior izquierda de su contenedor.
Text	Obtiene o establece el texto asociado al control.
Visible	Obtiene o establece un valor que indica si se muestran

	el control y todos sus controles primarios.
Tag	Obtiene o establece el objeto que contiene datos sobre el control.

MÉTODOS

Nombre	Descripción
Focus	Establece el foco de entrada en el control.
Show	Muestra el control al usuario.

2.- checkBox



CheckBox



checkBox1

Permite al usuario seleccionar o quitar la opción asociada.

PROPIEDADES

Nombre	Descripción
AllowDrop	Indica si el control puede aceptar datos que el usuario arrastre.
Anchor	Define a que bordes del contenedor está enlazado
AutoElipsis	Habilitar el tratamiento automatico del texto que se extiende mas alla del ancho del botón.
Autosize	Especifica si un control cambiara automáticamente de tamaño para ajustarse a su contenido.
BackColor	El color de fondo del componente.
CausesValidation	Indica si este componente genera eventos de validación.
CheckAling	Determina la ubicación de la casilla dentro del control.
FlatApperance	Para los botones cuyo FlatStyle es FlatStyle.Flat, determina la apariencia del borde y los colores utilizados para indicar el estado da activación el estado del mouse.
Dock	Define los bordes de control enlazados al contenedor.
Enabled	Indica si el control esta habilitado.
GenerateMember	Indica si se generará una variable miembro para este componente.
Image	Imagen que se mostrara en el control
Locked	La propiedad si se puede mover o cambiar el tamaño del control.
Margin	Especifica el espacio entre este control y el margen de otro control.
Size	Tamaño del control, en pixeles.
ThreeState	Indica si CheckBox permitirá tres estados de activación en lugares de dos.

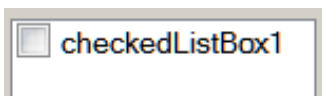
MÉTODOS

Nombre	Descripción	Nombre	Descripción
OnClick			Provoca el evento click.
Focus			Establece el foco de entrada en el control.
Show			Muestra el control al usuario.

3.- checkedListBox



CheckedListBox



Muestra una lista de elementos con una casilla a la izquierda de cada elemento.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
CheckOnClick			Indica si la casilla debe cambiar su estado al hacer clic por primera vez en un elemento.
ColumnWidth			Indica el ancho que debe tener las columnas de un control ListBox de varias columnas.
FormatString			Caracteres de especificador de formato que indica como se muestra un valor.
FormattingEnabled			Si esta propiedad es true, el valor de FormatString se utiliza para convertir el valor de DisplayMember en un valor que se pueda mostrar.
GenerateMember			Indica si se genera una variable miembro para este componente.
ImeMode			Determina el estado IME (editor de métodos de entrada) del objeto cuando se selecciona.
IntegralHeight			Indica si la lista puede contener solo elementos completos.
ScrollAlwaysVisible			Indica si el cuadro de lista debe tener siempre una barra de desplazamiento, independientemente de los elementos que tenga.
Sorted			Controla si la lista esta ordenada.
TabIndex			Determina el índice del orden de tabulación que ocupara este control.
ToolTip en tooltip1			Determina la información sobre herramientas que se muestra cuando se desplaza el puntero sobre el control.
TabStop			Indica si el usuario puede utilizar la tecla TABULADOR para poner el foco en el control.

MÉTODOS

Nombre	Descripción	Nombre	Descripción
OnClick			Provoca el evento click.
Focus			Establece el foco de entrada en el control.
Show			Muestra el control al usuario.

4.- ComboBox



ComboBox



Muestra un cuadro de texto editable con una lista desplegable de los valores permitidos.

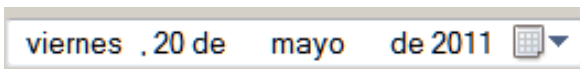
PROPIEDADES

Nombre	Descripción	Nombre	Descripción
AutoCompleteSource			El origen de cadenas completas utilizadas para la finalización automática.
AutoCompleteMode			Indica el comportamiento de finalización del texto del cuadro combinado.
Cursor			Cursos que aparece al pasar el puntero por el control.
DrawMode			Indica si el código o el sistema operativo controlara el dibujo de elementos de la lista.
DropDownHeight			Alto, en pixeles, del cuadro desplegable en un cuadro combinado.
DropDownStyle			Comprueba la apariencia y funcionalidad del cuadro combinado.
DropDownWidth			Ancho, en pixeles, del cuadrado desplegable en un cuadro combinado.
MaxLength			Especifica el número máximo de caracteres que puede introducirse en un cuadro combinado.
Modifiers			Indica el nivel de visibilidad del objeto.
Stored			Especifica si están ordenados los elementos en la lista del cuadro combinado.
UseWaitCursor			Cuando esta propiedad es true, la propiedad cursos del control y sus controles secundarios se establece en waitcursor.

5.-DateTimePicker



DateTimePicker



Permite al usuario seleccionar fecha y hora, así como mostrar ambas en un formato.

PROPIEDADES

Nombre Descripción	Nombre Descripción
CalendarFont	Fuente utilizada para mostrar el calendario.
CalendarMonthBackground	Color de fondo que se muestra dentro del mes.
CalendarTitleBackColor	Color de fondo que se muestra en el título del calendario.
CalendarTitleForeColor	Color utilizado para mostrar texto en el título del calendario.
DropDownAling	Controla la alineación del calendario desplegable con el selector de fecha DateTimePicker. Puede tener los valores LEFT o RIGHT.
GenerarMember	Indica si se genera una variable miembro para este componente.
MaxDate	Fecha máxima que se puede seleccionar.
MinDate	Fecha mínima que se puede seleccionar.
Value	Valor de fecha y hora actual para ese control.

MÉTODOS

Nombre Descripción	Nombre Descripción

6.-Label

A Label label1

Proporciona información en un tiempo de ejecución o texto descriptivo para un control.

PROPIEDADES

Nombre Descripción	Nombre Descripción
AutoEllipsis	Permite el control automático del texto que se extiende más allá del ancho del control de la etiqueta.
Padding	Especifica el espacio interior de un control.
RightToLeft	Indica si el componente debe dibujar de derecha a izquierda para idiomas RLT.
Tag	Datos definidos por el usuario asociados con el objeto.
UseMnemonic	Si es true, el primer carácter precedido por un símbolo de "y" comercial (&) se utiliza como la tecla de acceso de la etiqueta.

7.-LinkLabel



Muestra un control de etiqueta que admite funcionalidad de hipervínculo, formato y seguimiento.

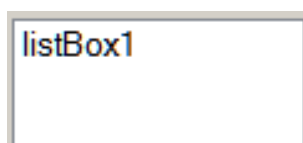
PROPIEDADES

Nombre	Descripción	Nombre	Descripción
activeLinkcolor			Determina el color del hipervínculo cuando el usuario hace clic en el vínculo.
DisableLinkColor			Determina el color del hipervínculo cuando esta deshabilitado.
LinkArea			Parte del texto de la etiqueta que se procesa como un hipervínculo.
LinkBehavior			Determina el comportamiento del subrayado de un hipervínculo.
LinkColor			Determina el color hipervínculo en el estado predeterminado.
LinkVisited			Determina si el hipervínculo debe representarse como visitado
VisitedLinkColor			Determine el color del hipervínculo cuando la propiedad LinkVisited esta establecida en true.

8.-ListBox



ListBox



Muestra una lista de la cual el usuario puede elegir elementos.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
IntegralHeight			Indica si la lista puede contener solo elementos completos.
Items			Elementos en el cuadro de lista
Margin			Especifica el espacio entre este control y el margen de otro control
MultiColumn			Indicca si los valores deben mostrarse horizontalmente por columnas.
Sorted			Controla si la lista esta ordenada.

Visible	Determina si el control esta visible u oculto.
---------	--

9.-ListView



Muestra una colección de elementos en una de 5 vistas diferentes.

PROPIEDADES

Nombre	Descripción
Activation	Indica el tipo de acción que debe realizar el usuario para activar un elemento y los comentarios proporcionados.
Alignment	Indica como están alineados los elementos en ListView.
Columns	Columnas mostrada en la vista de detalles.
Groups	Grupos de ListView.
MultiSelect	Permite seleccionar varios elementos.
OwnerDraw	Controla si el sistema o el usuario pinta los elementos o subelementos.
Scrollable	Indica si el control muestra las barras de desplazamiento porque contiene mas elementos de los que son posibles en el ares de clientes.
Sorting	Indica la manera en la que se ordenan los elementos.
TitleSize	Tamaño del mosaico en la vista de mosaico.
VirtualMode	Habilitar el modo virtual de ListView.

10. – MaskedTextBox



Utiliza una mascara para distinguir si los datos que especifica el usuario son apropiados o inapropiados.

PROPIEDADES

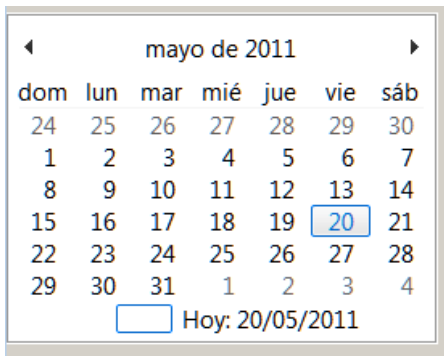
Nombre	Descripción
AsciiOnly	Indica si los caracteres ASCII se aceptan como entrada valida.
Culture	Referencia cultural que determina el valor de los

	deparadores y los marcadores de posición de idioma de la máscara localizable.
Mask	Establece la cadena que controla la entrada permitida para este control.
PasswordChar	Indica el carácter que se va a mostrar para la entrada de contraseña.
TextMaskFormat	Indica si la cadena devuelta de la propiedad Text incluye literales o caracteres de petición de datos.

11. – MonthCalendar



MonthCalendar



Muestra un calendario mensual del cual el usuario puede seleccionar una fecha.

PROPIEDADES

Nombre Descripción	Nombre Descripción
AnnuallyBodedDates	Indica que fechas anuales se deben poner en negrita.
CalendarDimensions	Numero de filas y columnas de mes en un calendario mensual.
FirstDayOfWeek	Primer día de la semana.
MaxDate	Fecha máxima que se puede seleccionar para un control de calendario mensual.
MiniDate	Fecha minima que se puede seleccionar para un control de calendario mensual.
TodayDate	Dia actual.

12. – NotifyIcon



NotifyIcon notifyIcon1

Muestra un icono en area de notificación, a la derecha de la barra de terea de Windows en tiempo de ejecución.

PROPIEDADES

Nombre Descripción	Nombre Descripción
--------------------	--------------------

BallonTipIcon	Icono que se va a asociar con la información sobre herramientas de globo.
BallonTipText	Texto que se va a asociar con la información sobre herramientas de globo
BallonTipTile	Titulo de la información sobre herramientas de globo.
ContextMenuStrip	El menú contextual que se mostrara cuando el usuario haga clic con el botón secundario en el icono.
GenerateMember	Indica si se genera una variable miembro para este componente.
Icon	Icono de la bandeja del sistema
Modifiers	Indica el nivel de visibilidad del objeto.
Tag	Datos definidos por el usuario asociados con el objeto.
Text	Texto que se muestra cuando se desplaza el puntero sobre el icono.
Visible	Determina si el control esta visible u oculto.

MÉTODOS

Nombre Descripción	Nombre Descripción

13. - NumericUpDown



Muestra un unico valor numeric que solo el usuario puede aumentar o reducir haciendo clic en los botones de arriba y abajo del control.

PROPIEDADES

Nombre Descripción	Nombre Descripción
DecimalPlaces	Indica el numero de posiciones decimal que se muestra.
Dock	Define los bordes del control enlazados al contenedor.
Hexadecimal	Indica si el control numérico de fichas debe mostrar el valor como hexadecimal.
Increment	Indica la cantidad que se va a aumentar o disminuir cada vez que hace clic en el botón.
Modifiers	Indica el nivel de visibilidad del objeto.
ReadOnly	Indica si el cuadro de edición es de solo lectura.
Value	Valor actual del control numérico de fichas.

14. - PictureBox



PictureBox



Muestra una imagen.

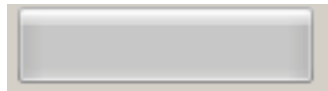
PROPIEDADES

Nombre	Descripción	Nombre	Descripción
ErrorImage		Imagen que se muestra cuando falla la carga de otra imagen.	
GenerateMember		Indica si se genera una variable miembro para este componente.	
Image		Imagen mostrada en el control PictureBox	
ImageLocation		Disco o ubicación web desde la que carga la imagen.	
WaitOnLoad		Controla si el procesamiento se detendrá hasta que la imagen se cargue.	

16. – ProgressBar



ProgressBar



Muestra una barra que se va completando para mostrar al usuario el progreso de una operación.

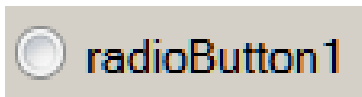
PROPIEDADES

Nombre	Descripción	Nombre	Descripción
Step		Cantidad por la que aumentar el valor actual del control cuando se emplea el método PerformStep().	
Style		Esta propiedad permite al usuario establece el estilo de progressbar.	
Value		Valor actual de progressbar, en el intervalo especificado por las propiedades de mínimo y máximo.	

17. – RadioButton



RadioButton



Permite al usuario seleccionar una única opción de entre una lista de opciones cuando están emparejadas con otros RadioButtons.

PROPIEDADES

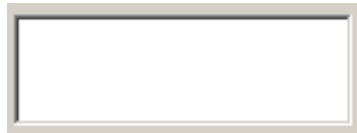
Nombre	Descripción	Nombre	Descripción
CheckAlign		Determina la ubicación de la casilla dentro del control.	

Checked	Indica si el botón de radio esta activado o no.
TextAlign	Alineación del texto que se mostrar en el control.

18. – RichTextBox



RichTextBox



Proporciona una entrada de texto y características de edición avanzada, como el formato de párrafo y caracteres.

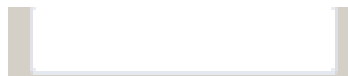
PROPIEDADES

Nombre	Descripción
AcceptsTab	Indica si se acepta caracteres de tabulación como entrada en controles de edición de múltiples líneas.
AutoWordSelection	Activa o desactiva la selección automática de palabras.
BulletIndent	Define la sangría de las viñetas en el control.
DetectUrls	Indica si las direcciones URL obtiene formato de vinculo automáticamente.
Multiline	Controla si el texto en el control de edición puede abarcar mas de una línea.
WordWrap	Indica si se realiza automáticamente el salto de líneas en controles de edición de varias líneas.
ZoomFactor	Defina el factor de escala actual de la presentación RichTextBox; 1.0 es la vista normal.

19. – TextBox



TextBox



Permite al usuario especificar texto, así como funciones de edición de varias líneas y mascarar caracteres para contraseña.

PROPIEDADES

Nombre	Descripción
AcceptsReturn	Indica si se aceptan los caracteres de retorno como entradas en los controles de edición de múltiples líneas.
AcceptsTab	Indica si se acepta caracteres de tabulación como entradas en controles de edición de múltiples líneas.
Lines	Líneas de texto de una edición de múltiples líneas, como una matriz de valores de cadena.
WordWrap	Indica si se realiza automáticamente el salto de líneas en controles de edición de varias líneas.

20. - ToolTip toolTip1

Muestra información cuando el usuario mueve el puntero sobre un control asociado.

PROPIEDADES

Nombre	Descripción
Active	Determina si la información sobre herramientas esta activada. Solo aparece una sugerencia si la información sobre herramientas se ha activado.
ForeColor	Color de primer plano del control de información sobre herramientas.
GenerateMember	Indica si se genera una variable miembro para este componente.
IsBallon	Indica si la información sobre herramientas tomara la forma de un globo.
Modifiers	Indica el nivel de visibilidad del objeto.
UseAnimation	Cuando se establece en true, se utiliza las animaciones si la información sobre herramientas se muestra o esta oculta.
UseFading	Cuando se establece en true, se utiliza en efecto de atenuación si las informaciones sobre herramientas se muestran o están ocultas.

21. – TreeView



Muestra al usuario una colección jerárquica de elementos con etiqueta, que opcionalmente puede contener una imagen.

PROPIEDADES

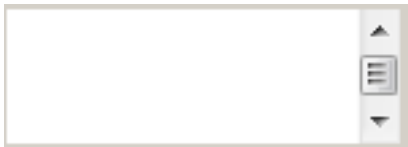
Nombre	Descripción
FullRowSelect	Indica si el resalte abarca al ancho de TreeView.
HotTracking	Indica si los nodos proporcionan comentarios cuando el mouse se mueve sobre ellos.
Indent	Ancho de sangría de los nodos secundarios en pixeles.
Nodes	Nodos raíz en el control TreeView.

StateImageList	Control ImageList utilizado por TreeView para estados personalizados.
----------------	---

22. – WebBrowser



WebBrowser

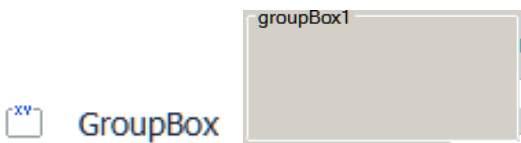


Permite al usuario explorar paginas web dentro del formulario.

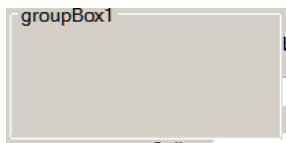
PROPIEDADES

Nombre	Descripción	Nombre	Descripción
AllowNavigation		Especifica si el control webbrowser puede desplazarse a otra pagina después de cargar inicialmente.	
Modifiers		Indica el nivel de visibilidad del objeto.	
Url		Especifica la dirección URL a la que ha navegado el explorador web	
WebBrowserShortcutsEnabled		Especifica si algunas teclas de aceleraciones están habilitadas en el control webbrowser.	

23. – GroupBox



GroupBox



Muestra un marco alrededor de un grupo de controles con un titulo opcional.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
Modifiers		Indica el nivel de visibilidad de objeto	
Text		Texto asociado al control	
FontColor		Color del primer plano de este componente, utilizado para mostrar texto.	

24. – Panel

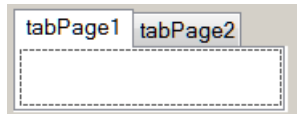
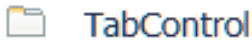


Permite agrupar colecciones de controles.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
AutoScrollMargin			Margen alrededor de los controles durante el desplazamiento automático.
AutoScrollminSize			Tamaño lógico mínimo para la zona de desplazamiento automático.
AccessibleRole			Función que se proporciona a clientes con discapacidades para facilitar el acceso.

25. – TabControl

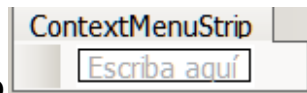
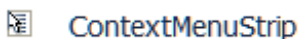


Controla y muestra al usuario una colección relacionada de fichas que puede contener controles y componentes.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
Alignment			Determina si las fichas aparecen en la parte superior, inferior izquierda o derecha del control.
Appearance			Indica si las fichas se representan como botones o como fichas normales.
TabIndex			Determina el índice del origen del orden de tabulación que ocupara este control.

26. – ContextMenuStrip



Muestra un menú contextual cuando el usuario hace clic sobre el botón secundario del mouse con el control asociado.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
--------	-------------	--------	-------------

AllowMerge	Permite combinar los elementos.
Margin	Especifica el estado entre este control y el margen de otro control.
TabStop	Indica si el usuario puede utilizar la tecla TABULADORA para poner el foco en el control.

27. – MenuStrip

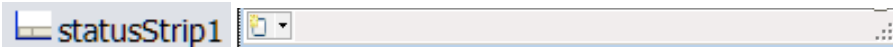


Muestra los comandos y las opciones de la aplicación agrupados por funcionalidad.

PROPIEDADES

Nombre Descripción	Nombre Descripción
AllowMerge	Permite combinar los elementos.
AutoSize	Especifica si un control cabiera de tamaño para ajustarse a su contenido.
TextDirection	especifica la dirección en la que dibujar el tecto en el elemento.

28. – StatusStrip

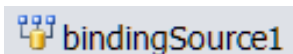


Muestra información para el usuario acerca de objeto que se esta viendo, los componentes del mismo o su funcionamiento.

PROPIEDADES

Nombre Descripción	Nombre Descripción
Locked	La propiedad locked determina si se puede mover o cambiar el tamaño del control.
SizingGrip	Determina si StatusStrip tiene un control de tamaño.
Size	Tamaño de control, en pixeles.

29. - BindingSource



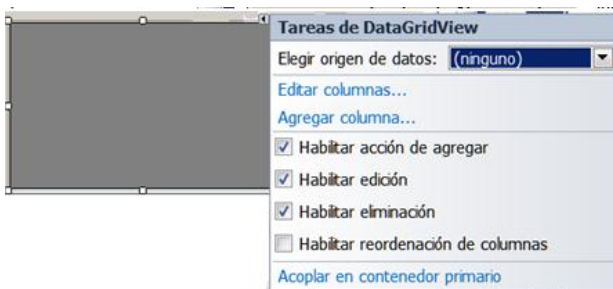
Encapsula un origen de datos de un formulario y proporciona funciones de navegación, filtrado, ordenación y actualización.

PROPIEDADES

Nombre	Descripción
AllowNew	Determina si DataSource permite agregar nuevos elementos a la lista.
DataMember	Indica una sublista de DataSource a lo que se encuentra enlazado DataSource.
DataSource	Indica el origen de los datos de DataSource.
Filter	Indica una expresión de columnas de base de datos utilizada para filtrar el conjunto de filas devuelto por el origen de datos.
GenerateMember	Indica si se genera una variable miembro para esta componente.
Modifiers	Indica el nivel de visibilidad del objeto.
Sort	Indica nombres de columnas de bases de datos utilizados para ordenar el conjunto de filas devuelto por el origen de datos.

30. - DataGridView

 DataGridView



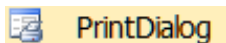
Muestra filas y columnas de datos en una cuadrícula que se puede personalizar.

PROPIEDADES

Nombre	Descripción
AllowUserToAddRows	Indica si la opción de agregar filas se muestra al usuario.
AllowUserToDeleteRows	Indica si el usuario puede eliminar filas de

	DataGridView.
AllowUserToOrderColumns	Indica si la reordenación manual de la columna esta habitada.
AllowUserToResizeColumns	Indica si el usuario puede cambiar el tamaño de las columnas.
DataSource	Indica el origen de datos para el control DataGridView.
EditMode	Identificar el modo que determina como se inicia la edición de las celdas.

32. – PrintDialog

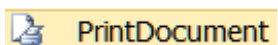


Muestra un cuadro de dialogo que permite al usuario seleccionar una impresora y elegir otras opciones de impresión, como el número de copias y la orientación de página.

PROPIEDADES

Nombre	Descripción
AllowCurrentPage	Habilita y deshabilitael botón de opción pagina actual.
Document	Donde se obtiene la configuración de impresora.
PrintToFile	Determina si la casill imprimir a un archivo esta activada.
ShowHelp	Controla si se muestra el botón ayuda.
ShowNetwork	Controla si se muestra el botón red.

33. - PrintDocument



Define un objeto que envía la salida una impresora.

PROPIEDADES

Nombre	Descripción
DocumentName	Nombre del documento mostrado al usuario
GenerateMember	Indica si se genera una variable miembro para este componente.
Modifiers	Indica el nivel de visibilidad del objeto

34. - PrintPreviewDialog



PrintPreviewDialog

Muestra un cuadro de dialogo que muestra al usuario la vista previa de cómo será el documento asociado cuando se imprima.

PROPIEDADES

Nombre	Descripción
AutoSizeMode	Especifica el modo mediante el que el elementos de interfaz del usuario cambia de tamaño automáticamente.
ShowIcon	Indica si un icono se muestra en la barra de titulo del formulario.
UseAntiAlias	Controla si la vista previa de impresión se procesara con suavizado de contorno.

35. – OpenFileDialog



OpenFileDialog

Muestra un cuadro de dialogo donde se pide al usuario que abra un archivo.

PROPIEDADES

Nombre	Descripción
AddExtension	Controla si la extensiones se agregan automáticamente a los nombres de archivo.
FileName	Primer achivo que se muestra en el cuadro de dialogo o el ultimo archivo seleccionado porel usuario.
Multiselect	Controla si se puede seleccionar varios archivos en el cuadro de dialogo.
ValidateName	Controla si el cuadro de dialogo comprueba que los nombres de archivos no contienen caracteres o secuencias no validas.

36. - SaveFileDialog



SaveFileDialog

Muestra un cuadro de dialogo donde se pide al usuario que seleccione una ubicación para guardar un archivo.

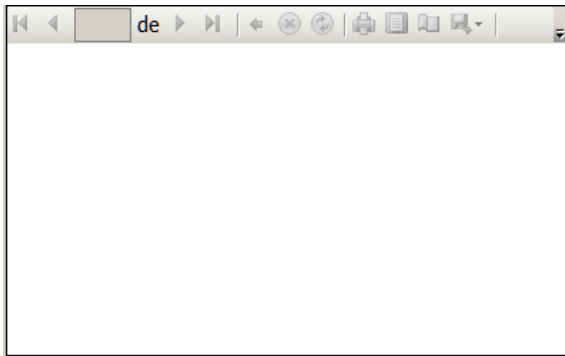
PROPIEDADES

Nombre	Descripción
CheckFileExists	Comprueba que el archivo especificado existe antes de volver del cuadro de dialogo.

Create Prompt	Controla si se debe avisar el usuario cuando se a crear nuevo archivo.
RestoreDirectory	Controla si el cuadro de dialogo restaura el directos actual antes de cerrar.

37. – ReportViewer

 ReportViewer



Muestra un informe.

PROPIEDADES

Nombre	Descripción	Nombre	Descripción
LocalReport	Propiedades de procesamiento local.		
PromptAreaCollapsed	Determina si el área de mensajes esta visible o contrida		
ZoomParcent	Determina el porcentaje de zoom aplicado al informe cuando ZoomMode está establecido como porcetaje.		

FUNDAMENTOS DEL LENGUAJE C

LENGUAJE C SHARP

C# es un lenguaje de programación orientado a objetos que utiliza clases y estructuras para implementar tipos como formularios Windows Forms, controles de interfaz de usuario y estructuras de datos. Una aplicación de C# típica se compone de clases definidas por el programador, junto con clases de .NET Framework.

C# ofrece varias maneras eficaces de definir clases, por ejemplo, proporcionar niveles de acceso diferentes, heredar características de otras clases y permitir que el programador especifique qué sucede cuando se crean o destruyen instancias de tipos.

CLASES

En C#, una clase es un tipo de datos muy eficaz. Como las estructuras, las clases definen los datos y el comportamiento del tipo de datos. Los programadores pueden crear objetos que son instancias de una clase. A diferencia de las estructuras, las clases admiten herencia, que es una parte fundamental de la programación orientada a objetos.

Las clases tienen las propiedades siguientes:

- Una clase puede implementar más de una interfaz.
- Las definiciones de clase se pueden dividir entre archivos de código fuente diferentes.
- Las clases estáticas son clases selladas que contienen sólo métodos estáticos.

Las clases se declaran mediante la palabra clave `class`, ejemplo siguiente:

```
public class Customer
{
    //Fields, properties, methods and events go here...
}
```

OBJETOS

Los objetos son construcciones de programación que se componen de datos, comportamiento e identidad. Los datos del objeto se encuentran en los campos, propiedades y eventos del objeto; los métodos e interfaces del objeto definen los comportamientos del objeto.

Los objetos tienen identidad, es decir, dos objetos con el mismo conjunto de datos no son necesariamente el mismo objeto.

En C#, los objetos se definen mediante **classes** y **structs**, que conforman el plano único a partir del cual operan todos los objetos de ese tipo.

Los objetos tienen las propiedades siguientes:

- Todo lo que se utiliza en C# es un objeto, incluidos los formularios Windows Forms y los controles.
- Se crean instancias de objetos; es decir, éstos se crean a partir de plantillas definidas por clases y estructuras.

- Los objetos utilizan propiedades para obtener y cambiar la información que contienen.
- A menudo, los objetos tienen métodos y eventos que les permiten realizar acciones.
- Visual Studio proporciona herramientas para manipular objetos: la Propiedades permite cambiar los atributos de los objetos, como formularios Windows Forms. El Examinador de objetos permite examinar el contenido de un objeto.
- Todos los objetos de C# heredan de Object.

Los objetos se pueden crear con la palabra clave `new` seguida del nombre de la clase en la que se basará el objeto, de la manera siguiente:

```
Customer object1 = new Customer();
```

VARIABLES

Una variable representa un valor numérico o de cadena o un objeto de una clase. El valor que la variable almacena puede cambiar, pero el nombre sigue siendo el mismo. Una variable es un tipo de *campo*. El código siguiente es un ejemplo sencillo de cómo declarar una variable de entero, asignarle un valor y, a continuación, asignarle un nuevo valor.

```
int x = 1; // x holds the value 1
x = 2;    // now x holds the value 2
```

DECLARACIÓN DE VARIABLES

Al declarar una variable o una constante en un programa, debe especificar su tipo o utilizar la palabra clave `var` para permitir que el compilador infiera el tipo

```
// Declaration only:
float temperature;
String name;
MyClass myClass;

// Declaration with initializers (four examples):
Char firstLetter = 'C';
Var limit = 3;
Int [] source = {0, 1, 2, 3, 4, 5};
Var query = from item in source
            Where item <= limit
            Select item;
```

CONSTANTES

Una constante es otro tipo de campo. Contiene un valor que se asigna cuando se compila el programa y nunca cambia después. Las constantes se declaran con la palabra clave `const`; son útiles para que el código sea más legible.

```
const int speedLimit = 55;  
const double pi = 3.14159265358979323846264338327950;
```

ATRIBUTOS

Los atributos constituyen un medio apropiado para asociar información declarativa con código de C# (tipos, métodos, propiedades, etc.). Una vez asociado a una entidad del programa, el atributo se puede consultar en tiempo de ejecución mediante una técnica denominada reflexión.

Existen dos formas de atributos: atributos que se definen en la biblioteca de clases base de Common Language Runtime y atributos personalizados que se pueden crear para agregar información adicional al código. Esta información se puede recuperar después mediante programación.

Los atributos tienen las propiedades siguientes:

- Los atributos agregan metadatos al programa. Los metadatos son información incrustada en el programa, tales como instrucciones del compilador o descripciones de datos.
- El programa puede examinar sus propios metadatos mediante la reflexión.
- Los atributos se utilizan normalmente al interactuar con COM.

INTERFACES

Las interfaces describen un grupo de comportamientos relacionados que pueden pertenecer a cualquier clase o estructura. Las interfaces pueden estar compuestas de métodos, propiedades, eventos, indizadores o cualquier combinación de estos cuatro tipos de miembros. Una interfaz no puede contener campos. Los miembros de interfaz son automáticamente públicos.

Una interfaz tiene las siguientes propiedades:

- Una interfaz es similar a una clase base abstracta. Cualquier tipo no abstracto que hereda la interfaz debe implementar todos sus miembros.
- No se pueden crear instancias directamente de una interfaz.

- Las interfaces pueden contener eventos, métodos, indizadores y propiedades.
- Las interfaces no contienen implementaciones de métodos.
- Las clases y estructuras se pueden heredar de más de una interfaz.
- Una interfaz se puede heredar de varias interfaces.

TIPOS DE DATOS

Un tipo de datos se puede describir como:

- Un tipo de datos integrado, como **int** o **char**, o bien
- Un tipo de datos definidos por el usuario, como **class** o **interface**.

Algunos tipos de datos te los presentamos a continuación:

Tipo	Intervalo	Tamaño
sbyte	-128 a 127	Entero de 8 bits con signo
byte	0 a 255	Entero de 8 bits sin signo
char	U+0000 a U+ffff	Carácter Unicode de 16 bits
short	-32.768 a 32.767	Entero de 16 bits con signo
ushort	0 a 65.535	Entero de 16 bits sin signo
int	-2.147.483.648 a 2.147.483.647	Entero de 32 bits con signo
uint	0 a 4.294.967.295	Entero de 32 bits sin signo
long	-9.223.372.036.854.775.808 a 9.223.372.036.854.775.807	Entero de 64 bits con signo
ulong	0 a 18.446.744.073.709.551.615	Entero de 64 bits sin signo

Tipo	Intervalo aproximado	Precisión
float	$\pm 1,5e-45$ a $\pm 3,4e38$	7 dígitos
decimal	$\pm 1,0 \times 10^{-28}$ a $\pm 7,9 \times 10^{28}$	28-29 dígitos significativos

double	$\pm 5,0e-324$ a $\pm 1,7e308$	15-16 dígitos
--------	--------------------------------	---------------

SENTENCIAS DE CONTROL

if

Al igual que en otros lenguajes de programación, tendremos varias posibilidades para comprobar una condición:

- comprobar una condición y en función de ésta realizar una acción
- comprobar una condición y en función de ésta realizar una acción u otra (excluyentes)
- comprobar varias condiciones para realizar varias acciones diferentes (excluyentes)

Esta estructura es muy parecida a la utilizada en C. La sintaxis de la instrucción if es la siguiente:

```
if (expresión) {
    instrucción o bloque de intrucciones 1;
}
[else {
    instrucción o bloque de intrucciones 2;
}]
```

switch

LA INSTRUCCION SWITCH puede ser de tipo Char, Byte, Short o Int (no se permiten instrucciones de punto flotante)

Esquema de un programa escrito en forma Switch

```
switch (expresion) {
case constante 1:
    secuencia de instrucciones;
    break;
case constante 2:
    secuencia de instrucciones;
    break;
.
.
.
.
```

```
default
secuencia de instrucciones;
}
```

while

La **instrucción while** permite ejecutar un bloque de instrucciones mientras se de una cierta instrucción. Su sintaxis de uso es:

```
while (<condición>)
{
    <instrucciones>
}
```

Su significado es el siguiente: Se evalúa la <condición> indicada, que ha de producir un valor lógico. Si es cierta (valor lógico **true**) se ejecutan las <instrucciones> y se repite el proceso de evaluación de <condición> y ejecución de <instrucciones> hasta que deje de serlo. Cuando sea falsa (**false**) se pasará a ejecutar la instrucción siguiente al **while**. En realidad <instrucciones> puede ser una única instrucción o un bloque de instrucciones.

```
using System;
class HolaMundoWhile
{
    public static void Main(String[] args)
    {
        int actual = 0;
        if (args.Length > 0)
            while (actual < args.Length)
            {
                Console.WriteLine(";Hola {0}!", args[actual]);
                actual = actual + 1;
            }
        else
            Console.WriteLine(";Hola mundo!");
    }
}
```

do...while

La instrucción **do...while** es una variante del **while** que se usa así:

```
do {
    <instrucciones>
} while(<condición>);
```

La única diferencia del significado de **do...while** respecto al de **while** es que en vez de evaluar primero la condición y ejecutar <instrucciones> sólo si es cierta, **do...while** primero ejecuta las <instrucciones> y luego mira la <condición> para ver si se ha de repetir la ejecución de las mismas. Por lo demás ambas instrucciones son iguales, e incluso también puede incluirse **break**; y **continue**; entre las <instrucciones> del **do...while**.

```
using System;
class HolaMundoDoWhile
{
    public static void Main()
    {
        String leído;
        do
        {
            Console.WriteLine("Clave: ");
            leído = Console.ReadLine();
        }
        while (leído != "José");
        Console.WriteLine("Hola José");
    }
}
```

for

La **instrucción for** es una variante de **while** que permite reducir el código necesario para escribir los tipos de bucles más comúnmente usados en programación. Su sintaxis es:

```
for (<inicialización>; <condición>; <modificación>){
    <instrucciones>
}
```

```
using System;
class HolaMundoFor
{
    public static void Main(String[] args)
    {
        if (args.Length > 0)
            for (int actual = 0; actual < args.Length; actual++) {
```

```
        Console.WriteLine(";Hola {0}!", args[actual]);
    }
    else
        Console.WriteLine(";Hola mundo!");
    }
}
```

PROPIEDADES (TODOS)

Propiedades públicas

Nombre	Descripción
<u>AccessibilityObject</u>	Obtiene AccessibleObject asignado al control.
<u>AccessibleDefaultActionDescription</u>	Obtiene o establece la descripción de la acción predeterminada del control que las aplicaciones cliente de accesibilidad utilizan.

<u>AccessibleDescription</u>	Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan.
<u>AccessibleName</u>	Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan.
<u>AccessibleRole</u>	Obtiene o establece la función accesible del control.
<u>AllowDrop</u>	Obtiene o establece un valor que indica si el control puede aceptar los datos que el usuario arrastra al mismo.
<u>Anchor</u>	Obtiene o establece los bordes del contenedor al que está enlazado un control y determina cómo se cambia el tamaño de un control con su elemento primario.
<u>AutoScrollOffset</u>	Obtiene o establece a dónde se desplaza este control en <u>ScrollControlIntoView</u> .
<u>AutoSize</u>	Esta propiedad no es relevante para esta clase.
<u>BackColor</u>	Obtiene o establece el color de fondo del control.
<u>BackgroundImage</u>	Obtiene o establece la imagen de fondo que se muestra en el control.
<u>BackgroundImageLayout</u>	Obtiene o establece el diseño de la imagen de fondo tal como se define en la enumeración <u>ImageLayout</u> .
<u>BindingContext</u>	Obtiene o establece BindingContext del control.
<u>Bottom</u>	Obtiene la distancia, en píxeles, que existe entre el borde inferior del control y el borde superior del área cliente de su contenedor.
<u>Bounds</u>	Obtiene o establece el tamaño y la ubicación del control incluyendo sus elementos no cliente, en píxeles, con respecto al control primario.
<u>CanFocus</u>	Obtiene un valor que indica si el control puede recibir el foco.
<u>CanSelect</u>	Obtiene un valor que indica si el control se puede seleccionar.
<u>Capture</u>	Obtiene o establece un valor que indica si el control ha capturado el mouse (ratón).

<u>CausesValidation</u>	Obtiene o establece un valor que indica si el control hace que se realice una validación de todos los controles que requieren validación cuando reciben el foco.
<u>CheckForIllegalCrossThreadCalls</u>	Obtiene o establece un valor que indica si se detectarán las llamadas en el subproceso equivocado que tiene acceso a la propiedad <u>Handle</u> de un control.
<u>ClientRectangle</u>	Obtiene el rectángulo que representa el área cliente del control.
<u>ClientSize</u>	Obtiene o establece el alto y el ancho del área cliente del control.
<u>CompanyName</u>	Obtiene el nombre de la compañía o del creador de la aplicación que contiene el control.
<u>Container</u>	Obtiene <u>IContainer</u> que contiene Component .(Se hereda de <u>Component</u>).
<u>ContainsFocus</u>	Obtiene un valor que indica si el control, o uno de sus controles secundarios, tiene el foco de entrada en la actualidad.
<u>ContextMenu</u>	Obtiene o establece el menú contextual asociado al control.
<u>ContextMenuStrip</u>	Obtiene o establece el <u>ContextMenuStrip</u> asociado a este control.
<u>Controls</u>	Obtiene la colección de controles que contiene el control.
<u>Created</u>	Obtiene un valor que indica si se ha creado el control.
<u>Cursor</u>	Obtiene o establece el cursor que se muestra cuando el puntero del mouse se sitúa sobre el control.
<u>DataBindings</u>	Obtiene los enlaces de datos del control.
<u>DefaultBackColor</u>	Obtiene el color de fondo predeterminado del control.
<u>DefaultFont</u>	Obtiene la fuente predeterminada del control.
<u>DefaultForeColor</u>	Obtiene el color de primer plano predeterminado del control.
<u>DisplayRectangle</u>	Obtiene el rectángulo que representa el área de presentación del

	control.
<u>Disposing</u>	Obtiene un valor que indica si la clase base Control está en el proceso de eliminación.
<u>Dock</u>	Obtiene o establece que los bordes del control se acoplarán a su control principal y determina cómo se cambia el tamaño de un control con su elemento primario.
<u>Enabled</u>	Obtiene o establece un valor que indica si el control puede responder a la interacción del usuario.
<u>Focused</u>	Obtiene un valor que indica si el control tiene el foco de entrada.
<u>Font</u>	Obtiene o establece la fuente del texto que muestra el control.
<u>ForeColor</u>	Obtiene o establece el color de primer plano del control.
<u>Handle</u>	Obtiene el identificador de ventana al que está enlazado el control.
<u>HasChildren</u>	Obtiene un valor que indica si el control contiene uno o más controles secundarios.
<u>Height</u>	Obtiene o establece el alto del control.
<u>ImeMode</u>	Obtiene o establece el modo de Editor de métodos de entrada (IME) del control.
<u>InvokeRequired</u>	Obtiene un valor que indica si el llamador debe llamar a un método de invocación cuando realiza llamadas a métodos del control porque el llamador se encuentra en un subproceso distinto al del control donde se creó.
<u>IsAccessible</u>	Obtiene o establece un valor que indica si el control es visible para las aplicaciones de accesibilidad.
<u>IsDisposed</u>	Obtiene un valor que indica si el control se ha eliminado.
<u>IsHandleCreated</u>	Obtiene un valor que indica si el control tiene un identificador asociado.
<u>IsMirrored</u>	Obtiene un valor que indica si el control está reflejado.

<u>LayoutEngine</u>	Obtiene una instancia almacenada en caché del motor de diseño del control.
<u>Left</u>	Obtiene o establece la distancia, en píxeles, que existe entre el borde izquierdo del control y el borde izquierdo del área cliente de su contenedor.
<u>Location</u>	Obtiene o establece las coordenadas de la esquina superior izquierda del control en relación con la esquina superior izquierda de su contenedor.
<u>Margin</u>	Obtiene o establece el espacio entre los controles.
<u>MaximumSize</u>	Obtiene o establece el tamaño que es el límite superior que <u>GetPreferredSize</u> puede especificar.
<u>MinimumSize</u>	Obtiene o establece el tamaño que es el límite inferior que GetPreferredSize puede especificar.
<u>ModifierKeys</u>	Obtiene un valor que indica cuál de las teclas modificadoras (MAYÚS, CTRL y ALT) está presionada.
<u>MouseButtons</u>	Obtiene un valor que indica cuál de los botones del mouse está presionado.
<u>MousePosition</u>	Obtiene la posición del cursor del mouse en coordenadas de pantalla.
<u>Name</u>	Obtiene o establece el nombre del control.
<u>Padding</u>	Obtiene o establece el relleno dentro del control.
<u>Parent</u>	Obtiene o establece el contenedor principal del control.
<u>PreferredSize</u>	Obtiene el tamaño de un área rectangular en la que cabe el control.
<u>ProductName</u>	Obtiene el nombre de producto del ensamblado que contiene el control.
<u>ProductVersion</u>	Obtiene la versión del ensamblado que contiene el control.
<u>RecreatingHandle</u>	Obtiene un valor que indica si el control está volviendo a crear

	su identificador en la actualidad.
<u>Region</u>	Obtiene o establece la región de ventana asociada al control.
<u>Right</u>	Obtiene la distancia, en píxeles, que existe entre el borde derecho del control y el borde izquierdo del área cliente de su contenedor.
<u>RightToLeft</u>	Obtiene o establece un valor que indica si los elementos del control se alinean para admitir configuraciones regionales utilizando fuentes de derecha a izquierda.
<u>Site</u>	Reemplazado. Obtiene o establece el sitio del control.
<u>Size</u>	Obtiene o establece el alto y el ancho del control.
<u>TabIndex</u>	Obtiene o establece el orden de tabulación del control en su contenedor.
<u>TabStop</u>	Obtiene o establece un valor que indica si el usuario puede dar el foco a este control mediante la tecla TAB.
<u>Tag</u>	Obtiene o establece el objeto que contiene datos sobre el control.
<u>Text</u>	Obtiene o establece el texto asociado al control.
<u>Top</u>	Obtiene o establece la distancia, en píxeles, que existe entre el borde superior del control y el borde superior del área cliente de su contenedor.
<u>TopLevelControl</u>	Obtiene el control principal que no es secundario de ningún otro control de formularios Windows Forms. Normalmente, se trata del <u>Form</u> más externo en el que está contenido el control.
<u>UseWaitCursor</u>	Obtiene o establece un valor que indica si se utiliza el cursor de espera para el control actual y todos los controles secundarios.
<u>Visible</u>	Obtiene o establece un valor que indica si se muestran el control y todos sus controles primarios.
<u>Width</u>	Obtiene o establece el ancho del control.

<u>WindowTarget</u>	Esta propiedad no es relevante para esta clase.
---------------------	---

Propiedades protegidas

Nombre	Descripción
<u>CanRaiseEvents</u>	Reemplazado. Determina si se pueden producir eventos en el control.
<u>CreateParams</u>	Obtiene los parámetros de creación necesarios cuando se crea el identificador del control.
<u>DefaultCursor</u>	Obtiene o establece el cursor predeterminado del control.
<u>DefaultImeMode</u>	Obtiene el modo de Editor de métodos de entrada (IME) predeterminado que admite el control.
<u>DefaultMargin</u>	Obtiene el espacio, en píxeles, que se especifica de manera predeterminada entre los controles.
<u>DefaultMaximumSize</u>	Obtiene la longitud y el alto, en píxeles, que se especifica como el tamaño máximo predeterminado de un control.
<u>DefaultMinimumSize</u>	Obtiene la longitud y el alto, en píxeles, que se especifica como el tamaño mínimo predeterminado de un control.
<u>DefaultPadding</u>	Obtiene el espaciado interno, en píxeles, del contenido de un control.
<u>DefaultSize</u>	Obtiene el tamaño predeterminado del control.
<u>DesignMode</u>	Obtiene un valor que indica si Component está actualmente en modo de diseño.(Se hereda de <u>Component</u>).
<u>DoubleBuffered</u>	Obtiene o establece un valor que indica si este control debe volver a dibujar su superficie mediante un búfer secundario para reducir o evitar el parpadeo.
<u>Events</u>	Obtiene la lista de controladores de eventos asociados a Component .(Se hereda de <u>Component</u>).
<u>FontHeight</u>	Obtiene o establece el alto de la fuente del control.
<u>RenderRightToLeft</u>	Esta propiedad está obsoleta.

<u>ResizeRedraw</u>	Obtiene o establece un valor que indica si el control vuelve a dibujarse automáticamente cuando cambia de tamaño.
<u>ScaleChildren</u>	Obtiene un valor que determina el escalado de los controles secundarios.
<u>ShowFocusCues</u>	Obtiene un valor que indica si el control debe mostrar rectángulos de foco.
<u>ShowKeyboardCues</u>	Obtiene un valor que indica si la interfaz de usuario está en el estado adecuado para mostrar u ocultar los aceleradores de teclado.

METODOS (TODOS)

Métodos públicos

Nombre	Descripción
<u>BeginInvoke</u>	Sobrecargado. Ejecuta un delegado de forma asíncrona en el subproceso donde se creó el identificador subyacente del control.
<u>BringToFront</u>	Coloca el control al principio del orden Z.
<u>Contains</u>	Obtiene un valor que indica si el control especificado es un control secundario del control.
<u>CreateControl</u>	Obliga a que se cree el control, incluidos el identificador y los controles secundarios.
<u>CreateGraphics</u>	Crea el <u>Graphics</u> para el control.
<u>CreateObjRef</u>	Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. (Se hereda de <u>MarshalByRefObject</u>).
<u>Dispose</u>	Sobrecargado. Libera todos los recursos utilizados por <u>Control</u> .
<u>DoDragDrop</u>	Inicia una operación de arrastrar y colocar.
<u>DrawToBitmap</u>	Permite la representación del mapa de bits especificado.
<u>EndInvoke</u>	Recupera el valor devuelto por la operación asíncrona representada por la <u>IAsyncResult</u> que se pasa.

<u>Equals</u>	Sobrecargado. Determina si dos instancias de <u>Object</u> son iguales. (Se hereda de <u>Object</u>).
<u>FindForm</u>	Recupera el formulario en el que se encuentra el control.
<u>Focus</u>	Establece el foco de entrada en el control.
<u>FromChildHandle</u>	Recupera el control que contiene el identificador especificado.
<u>FromHandle</u>	Devuelve el control actualmente asociado al identificador especificado.
<u>GetChildAtPoint</u>	Sobrecargado. Recupera el control secundario en una ubicación especificada.
<u>GetContainerControl</u>	Devuelve el siguiente <u>ContainerControl</u> (en sentido ascendente) de la cadena de controles principales del control.
<u>GetHashCode</u>	Actúa como función hash para un tipo concreto. (Se hereda de <u>Object</u>).
<u>GetLifetimeService</u>	Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia. (Se hereda de <u>MarshalByRefObject</u>).
<u>GetNextControl</u>	Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios.
<u>GetPreferredSize</u>	Recupera el tamaño de un área rectangular en la que puede caber un control.
<u>GetType</u>	Obtiene el objeto <u>Type</u> de la instancia actual. (Se hereda de <u>Object</u>).
<u>Hide</u>	Oculto el control al usuario.
<u>InitializeLifetimeService</u>	Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia. (Se hereda de <u>MarshalByRefObject</u>).
<u>Invalidate</u>	Sobrecargado. Invalida una región específica del control y hace que se envíe un mensaje de dibujo al control.
<u>Invoke</u>	Sobrecargado. Ejecuta un delegado en el subproceso que posee el identificador de ventana subyacente del control.
<u>IsKeyLocked</u>	Determina si la tecla BLOQ MAYÚS, BLOQ NUM o BLOQ DESPL está en

	vigor.
<u>IsMnemonic</u>	Determina si el carácter especificado es una tecla de acceso asignada al control en la cadena especificada.
<u>PerformLayout</u>	Sobrecargado. Obliga al control a aplicar la lógica de diseño a los controles secundarios.
<u>PointToClient</u>	Calcula la ubicación del punto especificado de la pantalla, en coordenadas de cliente.
<u>PointToScreen</u>	Calcula la ubicación del punto especificado de cliente en coordenadas de pantalla.
<u>PreProcessControlMessage</u>	Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos.
<u>PreProcessMessage</u>	Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos.
<u>RectangleToClient</u>	Calcula el tamaño y la ubicación del rectángulo de pantalla especificado, en coordenadas de cliente.
<u>RectangleToScreen</u>	Calcula el tamaño y la ubicación del rectángulo de cliente especificado, en coordenadas de pantalla.
<u>ReferenceEquals</u>	Determina si las instancias de Object especificadas son la misma instancia. (Se hereda de <u>Object</u>).
<u>Refresh</u>	Obliga al control a invalidar su área cliente y, acto seguido, obliga a que vuelva a dibujarse el control y sus controles secundarios.
<u>ResetBackColor</u>	Restablece el valor predeterminado de la propiedad <u>BackColor</u> .
<u>ResetBindings</u>	Hace que un control enlazado con <u>BindingSource</u> vuelva a leer todos los elementos de la lista y actualice los valores mostrados.
<u>ResetCursor</u>	Restablece el valor predeterminado de la propiedad <u>Cursor</u> .
<u>ResetFont</u>	Restablece el valor predeterminado de la propiedad <u>Font</u> .
<u>ResetForeColor</u>	Restablece el valor predeterminado de la propiedad <u>ForeColor</u> .

<u>ResetImeMode</u>	Restablece el valor predeterminado de la propiedad <u>ImeMode</u> .
<u>ResetRightToLeft</u>	Restablece el valor predeterminado de la propiedad <u>RightToLeft</u> .
<u>ResetText</u>	Restablece el valor predeterminado de la propiedad <u>Text</u> .
<u>ResumeLayout</u>	Sobrecargado. Reanuda la lógica de diseño habitual.
<u>Scale</u>	Sobrecargado. Ajusta la escala del control y de todos los controles secundarios.
<u>Select</u>	Sobrecargado. Activa un control.
<u>SelectNextControl</u>	Activa el siguiente control.
<u>SendToBack</u>	Envía el control al final del orden Z.
<u>SetBounds</u>	Sobrecargado. Establece los límites del control.
<u>Show</u>	Muestra el control al usuario.
<u>SuspendLayout</u>	Suspende temporalmente la lógica de diseño del control.
<u>ToString</u>	Devuelve un objeto <u>String</u> que contiene el nombre del objeto <u>Component</u> , en caso de que exista. Este método no debe reemplazarse. (Se hereda de <u>Component</u>).
<u>Update</u>	Hace que el control vuelva a dibujar las regiones no válidas en su área de cliente.

Métodos protegidos

Nombre	Descripción
<u>AccessibilityNotifyClients</u>	Sobrecargado. Notifica los objetos <u>AccessibleEvents</u> a las aplicaciones cliente de accesibilidad.
<u>CreateAccessibilityInstance</u>	Crea un nuevo objeto de accesibilidad para el control.
<u>CreateControlsInstance</u>	Crea una nueva instancia de la colección de controles para el control.
<u>CreateHandle</u>	Crea un identificador para el control.

<u>DefWndProc</u>	Envía el mensaje especificado al procedimiento de ventana predeterminado.
<u>DestroyHandle</u>	Destruye el identificador asociado a este control.
<u>Dispose</u>	Sobrecargado. Reemplazado. Libera todos los recursos utilizados por Control .
<u>Finalize</u>	Libera recursos no administrados y realiza otras operaciones de limpieza antes de que se reclame el objeto Component durante la recolección de elementos no utilizados. (Se hereda de <u>Component</u>).
<u>GetAccessibilityObjectById</u>	Recupera el <u>AccessibleObject</u> especificado.
<u>GetAutoSizeMode</u>	Recupera un valor que indica cómo se comportará un control cuando su propiedad <u>AutoSize</u> esté habilitada.
<u>GetScaledBounds</u>	Recupera los límites dentro de los cuales se escala el control.
<u>GetService</u>	Devuelve un objeto que representa el servicio suministrado por Component o por <u>Container</u> . (Se hereda de <u>Component</u>).
<u>GetStyle</u>	Recupera el valor del bit de estilo de control especificado para el control.
<u>GetTopLevel</u>	Determina si el control es de nivel superior.
<u>InitLayout</u>	Se llama a este método cuando el control se ha agregado a otro contenedor.
<u>InvokeGotFocus</u>	Provoca el evento <u>GotFocus</u> para el control especificado.
<u>InvokeLostFocus</u>	Provoca el evento <u>LostFocus</u> para el control especificado.
<u>InvokeOnClick</u>	Provoca el evento <u>Click</u> para el control especificado.
<u>InvokePaint</u>	Provoca el evento <u>Paint</u> para el control especificado.
<u>InvokePaintBackground</u>	Provoca el evento PaintBackground para el control especificado.
<u>IsInputChar</u>	Determina si un carácter es un carácter de entrada que el

	control reconoce.
<u>IsInputKey</u>	Determina si la tecla especificada es una tecla de entrada normal o una tecla especial que requiere preprocesamiento.
<u>MemberwiseClone</u>	Sobrecargado. (Se hereda de <u>MarshalByRefObject</u>).
<u>NotifyInvalidate</u>	Provoca el evento <u>Invalidated</u> con una región especificada del control para invalidar.
<u>OnAutoSizeChanged</u>	Provoca el evento <u>AutoSizeChanged</u> .
<u>OnBackColorChanged</u>	Provoca el evento <u>BackColorChanged</u> .
<u>OnBackgroundImageChanged</u>	Provoca el evento <u>BackgroundImageChanged</u> .
<u>OnBackgroundImageLayoutChanged</u>	Provoca el evento <u>BackgroundImageLayoutChanged</u> .
<u>OnBindingContextChanged</u>	Provoca el evento <u>BindingContextChanged</u> .
<u>OnCausesValidationChanged</u>	Provoca el evento <u>CausesValidationChanged</u> .
<u>OnChangeUICues</u>	Provoca el evento <u>ChangeUICues</u> .
<u>OnClick</u>	Provoca el evento Click .
<u>OnClientSizeChanged</u>	Provoca el evento <u>ClientSizeChanged</u> .
<u>OnContextMenuChanged</u>	Provoca el evento <u>ContextMenuChanged</u> .
<u>OnContextMenuStripChanged</u>	Provoca el evento <u>ContextMenuStripChanged</u> .
<u>OnControlAdded</u>	Provoca el evento <u>ControlAdded</u> .
<u>OnControlRemoved</u>	Provoca el evento <u>ControlRemoved</u> .
<u>OnCreateControl</u>	Provoca el método <u>CreateControl</u> .
<u>OnCursorChanged</u>	Provoca el evento <u>CursorChanged</u> .
<u>OnDockChanged</u>	Provoca el evento <u>DockChanged</u> .
<u>OnDoubleClick</u>	Provoca el evento <u>DoubleClick</u> .
<u>OnDragDrop</u>	Provoca el evento <u>DragDrop</u> .

<u>OnDragEnter</u>	Provoca el evento <u>DragEnter</u> .
<u>OnDragLeave</u>	Provoca el evento <u>DragLeave</u> .
<u>OnDragOver</u>	Provoca el evento <u>DragOver</u> .
<u>OnEnabledChanged</u>	Provoca el evento <u>EnabledChanged</u> .
<u>OnEnter</u>	Provoca el evento <u>Enter</u> .
<u>OnFontChanged</u>	Provoca el evento <u>FontChanged</u> .
<u>OnForeColorChanged</u>	Provoca el evento <u>ForeColorChanged</u> .
<u>OnGiveFeedback</u>	Provoca el evento <u>GiveFeedback</u> .
<u>OnGotFocus</u>	Provoca el evento GotFocus .
<u>OnHandleCreated</u>	Provoca el evento <u>HandleCreated</u> .
<u>OnHandleDestroyed</u>	Provoca el evento <u>HandleDestroyed</u> .
<u>OnHelpRequested</u>	Provoca el evento <u>HelpRequested</u> .
<u>OnImeModeChanged</u>	Provoca el evento <u>ImeModeChanged</u> .
<u>OnInvalidated</u>	Provoca el evento Invalidated .
<u>OnKeyDown</u>	Provoca el evento <u>KeyDown</u> .
<u>OnKeyPress</u>	Provoca el evento <u>KeyPress</u> .
<u>OnKeyUp</u>	Provoca el evento <u>KeyUp</u> .
<u>OnLayout</u>	Provoca el evento <u>Layout</u> .
<u>OnLeave</u>	Provoca el evento <u>Leave</u> .
<u>OnLocationChanged</u>	Provoca el evento <u>LocationChanged</u> .
<u>OnLostFocus</u>	Provoca el evento LostFocus .
<u>OnMarginChanged</u>	Provoca el evento <u>MarginChanged</u> .
<u>OnMouseCaptureChanged</u>	Provoca el evento <u>MouseCaptureChanged</u> .
<u>OnMouseClicked</u>	Provoca el evento <u>MouseClicked</u> .






<u>OnMouseDoubleClick</u>	Provoca el evento <u>MouseDoubleClick</u> .
<u>OnMouseDown</u>	Provoca el evento <u>MouseDown</u> .
<u>OnMouseEnter</u>	Provoca el evento <u>MouseEnter</u> .
<u>OnMouseHover</u>	Provoca el evento <u>MouseHover</u> .
<u>OnMouseLeave</u>	Provoca el evento <u>MouseLeave</u> .
<u>OnMouseMove</u>	Provoca el evento <u>MouseMove</u> .
<u>OnMouseUp</u>	Provoca el evento <u>MouseUp</u> .
<u>OnMouseWheel</u>	Provoca el evento <u>MouseWheel</u> .
<u>OnMove</u>	Provoca el evento <u>Move</u> .
<u>OnNotifyMessage</u>	Notifica al control los mensajes de Windows.
<u>OnPaddingChanged</u>	Provoca el evento <u>PaddingChanged</u> .
<u>OnPaint</u>	Provoca el evento Paint .
<u>OnPaintBackground</u>	Pinta el fondo del control.
<u>OnParentBackColorChanged</u>	Provoca el evento BackColorChanged cuando cambia el valor de la propiedad BackColor del contenedor del control.
<u>OnParentBackgroundImageChanged</u>	Provoca el evento BackgroundImageChanged cuando cambia el valor de la propiedad <u>BackgroundImage</u> del contenedor del control.
<u>OnParentBindingContextChanged</u>	Provoca el evento BindingContextChanged cuando cambia el valor de la propiedad <u>BindingContext</u> del contenedor del control.
<u>OnParentChanged</u>	Provoca el evento <u>ParentChanged</u> .
<u>OnParentCursorChanged</u>	Provoca el evento CursorChanged .
<u>OnParentEnabledChanged</u>	Provoca el evento EnabledChanged cuando cambia el valor de la propiedad <u>Enabled</u> del contenedor del control.
<u>OnParentFontChanged</u>	Provoca el evento FontChanged cuando cambia el valor de la

	propiedad Font del contenedor del control.
<u>OnParentForeColorChanged</u>	Provoca el evento ForeColorChanged cuando cambia el valor de la propiedad ForeColor del contenedor del control.
<u>OnParentRightToLeftChanged</u>	Provoca el evento <u>RightToLeftChanged</u> cuando cambia el valor de la propiedad RightToLeft del contenedor del control.
<u>OnParentVisibleChanged</u>	Provoca el evento <u>VisibleChanged</u> cuando cambia el valor de la propiedad <u>Visible</u> del contenedor del control.
<u>OnPreviewKeyDown</u>	Provoca el evento <u>PreviewKeyDown</u> .
<u>OnPrint</u>	Provoca el evento Paint .
<u>OnQueryContinueDrag</u>	Provoca el evento <u>QueryContinueDrag</u> .
<u>OnRegionChanged</u>	Provoca el evento <u>RegionChanged</u> .
<u>OnResize</u>	Provoca el evento <u>Resize</u> .
<u>OnRightToLeftChanged</u>	Provoca el evento RightToLeftChanged .
<u>OnSizeChanged</u>	Provoca el evento <u>SizeChanged</u> .
<u>OnStyleChanged</u>	Provoca el evento <u>StyleChanged</u> .
<u>OnSystemColorsChanged</u>	Provoca el evento <u>SystemColorsChanged</u> .
<u>OnTabIndexChanged</u>	Provoca el evento <u>TabIndexChanged</u> .
<u>OnTabStopChanged</u>	Provoca el evento <u>TabStopChanged</u> .
<u>OnTextChanged</u>	Provoca el evento <u>TextChanged</u> .
<u>OnValidated</u>	Provoca el evento <u>Validated</u> .
<u>OnValidating</u>	Provoca el evento <u>Validating</u> .
<u>OnVisibleChanged</u>	Provoca el evento VisibleChanged .
<u>ProcessCmdKey</u>	Procesa una tecla de comando.
<u>ProcessDialogChar</u>	Procesa un carácter de cuadro de diálogo.
<u>ProcessDialogKey</u>	Procesa una tecla de cuadro de diálogo.


















<u>ProcessKeyEventArgs</u>	Procesa un mensaje de tecla y genera los eventos de control correspondientes.
<u>ProcessKeyMessage</u>	Procesa un mensaje de teclado.
<u>ProcessKeyPreview</u>	Muestra una vista preliminar de un mensaje del teclado.
<u>ProcessMnemonic</u>	Procesa un carácter de tecla de acceso.
<u>RaiseDragEvent</u>	Provoca el evento de arrastre adecuado.
<u>RaiseKeyEvent</u>	Provoca el evento de tecla adecuado.
<u>RaiseMouseEvent</u>	Provoca el evento del mouse adecuado.
<u>RaisePaintEvent</u>	Provoca el evento de dibujo adecuado.
<u>RecreateHandle</u>	Obliga a que se vuelva a crear el identificador del control.
<u>ReflectMessage</u>	Refleja el mensaje especificado en el control que está enlazado al identificador especificado.
<u>ResetMouseEventArgs</u>	Restablece el control para controlar el evento MouseLeave .
<u>RtlTranslateAlignment</u>	Sobrecargado. Convierte la alineación actual en la alineación correspondiente para que admita texto de derecha a izquierda.
<u>RtlTranslateContent</u>	Convierte la <u>ContentAlignment</u> especificada en la ContentAlignment correspondiente para que admita texto de derecha a izquierda.
<u>RtlTranslateHorizontal</u>	Convierte la <u>HorizontalAlignment</u> especificada en la HorizontalAlignment correspondiente para que admita texto de derecha a izquierda.
<u>RtlTranslateLeftRight</u>	Convierte la <u>LeftRightAlignment</u> especificada en la LeftRightAlignment correspondiente para que admita texto de derecha a izquierda.
<u>ScaleControl</u>	Escala la ubicación, el tamaño, el relleno y el margen de un control.
<u>ScaleCore</u>	Este método no es relevante para esta clase.

<u>Select</u>	Sobrecargado. Activa un control.
<u>SetAutoSizeMode</u>	Establece un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada.
<u>SetBoundsCore</u>	Realiza la tarea de configurar los límites especificados de este control.
<u>SetClientSizeCore</u>	Establece el tamaño del área cliente del control.
<u>SetStyle</u>	Establece el bit de estilo especificado en el valor especificado.
<u>SetTopLevel</u>	Establece el control como el control de nivel superior.
<u>SetVisibleCore</u>	Establece el control en el estado de visibilidad especificado.
<u>SizeFromClientSize</u>	Determina el tamaño de todo el control a partir del alto y el ancho de su área cliente.
<u>UpdateBounds</u>	Sobrecargado. Actualiza los límites del control.
<u>UpdateStyles</u>	Obliga a que los estilos asignados vuelvan a aplicarse al control.
<u>UpdateZOrder</u>	Actualiza el control en el orden Z de su control principal.
<u>WndProc</u>	Procesa los mensajes de Windows.

EVENTOS

Nombre	Descripción
 <u>AutoSizeChanged</u>	Este evento no es relevante para esta clase.
 <u>BackColorChanged</u>	Se produce cuando cambia el valor de la propiedad <u>BackColor</u> .
 <u>BackgroundImageChanged</u>	Se produce cuando cambia el valor de la propiedad <u>BackgroundImage</u> .
 <u>BackgroundImageLayoutChanged</u>	Se produce cuando cambia la propiedad <u>BackgroundImageLayout</u> .
 <u>BindingContextChanged</u>	Se produce cuando cambia el valor de la propiedad

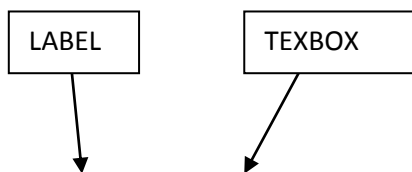
		<u>BindingContext</u> .
⚡	<u>CausesValidationChanged</u>	Se produce cuando cambia el valor de la propiedad <u>CausesValidation</u> .
⚡	<u>ChangeUICues</u>	Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado.
⚡	<u>Click</u>	Se produce cuando se hace clic en el control.
⚡	<u>ClientSizeChanged</u>	Se produce cuando cambia el valor de la propiedad <u>ClientSize</u> .
⚡	<u>ContextMenuChanged</u>	Se produce cuando cambia el valor de la propiedad <u>ContextMenu</u> .
⚡	<u>ContextMenuStripChanged</u>	Se produce cuando cambia el valor de la propiedad <u>ContextMenuStrip</u> .
⚡	<u>ControlAdded</u>	Se produce cuando se agrega un nuevo control a <u>Control.ControlCollection</u> .
⚡	<u>ControlRemoved</u>	Se produce cuando se quita un control de Control.ControlCollection .
⚡	<u>CursorChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Cursor</u> .
⚡	<u>Disposed</u>	Se produce cuando el componente se elimina mediante una llamada al método <u>Dispose</u> . (Se hereda de <u>Component</u>).
⚡	<u>DockChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Dock</u> .
⚡	<u>DoubleClick</u>	Se produce cuando se hace doble clic en el control.
⚡	<u>DragDrop</u>	Se produce cuando termina una operación de arrastrar y colocar.
⚡	<u>DragEnter</u>	Se produce cuando se arrastra un objeto dentro de los límites del control.
⚡	<u>DragLeave</u>	Se produce cuando se arrastra un objeto fuera de los límites del control.
⚡	<u>DragOver</u>	Se produce cuando se arrastra un objeto sobre los límites del

		control.
	<u>EnabledChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Enabled</u> .
	<u>Enter</u>	Se produce cuando se entra en el control.
	<u>FontChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Font</u> .
	<u>ForeColorChanged</u>	Se produce cuando cambia el valor de la propiedad <u>ForeColor</u> .
	<u>GiveFeedback</u>	Se produce durante una operación de arrastre.
	<u>GotFocus</u>	Se produce cuando el control recibe el foco.
	<u>HandleCreated</u>	Se produce cuando se crea un identificador para el control.
	<u>HandleDestroyed</u>	Se produce cuando el identificador del control está en proceso de eliminación.
	<u>HelpRequested</u>	Se produce cuando el usuario solicita ayuda para un control.
	<u>ImeModeChanged</u>	Se produce cuando ha cambiado la propiedad <u>ImeMode</u> .
	<u>Invalidated</u>	Se produce cuando es necesario volver a dibujar un control.
	<u>KeyDown</u>	Se produce cuando se presiona una tecla mientras el control tiene el foco.
	<u>KeyPress</u>	Se produce cuando se presiona una tecla mientras el control tiene el foco.
	<u>KeyUp</u>	Se produce cuando se suelta una tecla mientras el control tiene el foco.
	<u>Layout</u>	Se produce cuando un control debe volver a colocar sus controles secundarios.
	<u>Leave</u>	Se produce cuando el foco de entrada deja el control.
	<u>LocationChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Location</u> .
	<u>LostFocus</u>	Se produce cuando el control pierde el foco.
	<u>MarginChanged</u>	Se produce cuando cambia el margen del control.

 <u>MouseCaptureChanged</u>	Se produce cuando el control pierde o gana la captura del mouse.
 <u>MouseClicked</u>	Se produce cuando se hace clic en el control con el mouse.
 <u>MouseDoubleClick</u>	Se produce cuando se hace doble clic en el control con el mouse.
 <u>MouseDown</u>	Se produce cuando el puntero del mouse (ratón) se encuentra sobre el control y se presiona un botón del mouse.
 <u>MouseEnter</u>	Se produce cuando el puntero del mouse entra en el control.
 <u>MouseHover</u>	Se produce cuando el puntero del mouse se sitúa sobre el control.
 <u>MouseLeave</u>	Se produce cuando el puntero del mouse deja el control.
 <u>MouseMove</u>	Se produce cuando el puntero del mouse se mueve sobre el control.
 <u>MouseUp</u>	Se produce cuando el puntero del mouse se encuentra sobre el control y se suelta un botón del mouse.
 <u>MouseWheel</u>	Se produce cuando la rueda del mouse se mueve mientras el control tiene el foco.
 <u>Move</u>	Se produce cuando se mueve el control.
 <u>PaddingChanged</u>	Se produce cuando cambia el relleno del control.
 <u>Paint</u>	Se produce cuando vuelve a dibujarse el control.
 <u>ParentChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Parent</u> .
 <u>PreviewKeyDown</u>	Se produce antes que el evento <u>KeyDown</u> cuando se presiona una tecla mientras el foco está en este control.
 <u>QueryAccessibilityHelp</u>	Se produce cuando <u>AccessibleObject</u> proporciona ayuda para aplicaciones de accesibilidad.
 <u>QueryContinueDrag</u>	Se produce durante una operación de arrastrar y colocar, y permite al origen de arrastre determinar si la operación de

		arrastrar y colocar tiene que cancelarse.
⚡	<u>RegionChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Region</u> .
⚡	<u>Resize</u>	Se produce cuando se cambia el tamaño del control.
⚡	<u>RightToLeftChanged</u>	Se produce cuando cambia el valor de la propiedad <u>RightToLeft</u> .
⚡	<u>SizeChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Size</u> .
⚡	<u>StyleChanged</u>	Se produce cuando cambia el estilo del control.
⚡	<u>SystemColorsChanged</u>	Se produce cuando se modifican los colores del sistema.
⚡	<u>TabIndexChanged</u>	Se produce cuando cambia el valor de la propiedad <u>TabIndex</u> .
⚡	<u>TabStopChanged</u>	Se produce cuando cambia el valor de la propiedad <u>TabStop</u> .
⚡	<u>TextChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Text</u> .
⚡	<u>Validated</u>	Se produce cuando finaliza la validación del control.
⚡	<u>Validating</u>	Se produce cuando el control se está validando.
⚡	<u>VisibleChanged</u>	Se produce cuando cambia el valor de la propiedad <u>Visible</u> .

PANTALLA DE EJEMPLO:



punto_venta

Datos Generales De la Venta

ID Fecha Cliente

Detalles de la venta

Producto

Lista de productos

	Clave	Nombre	PrecioUnitario	Descipcion
*	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Datos del cobro

Su pago Total \$0.00 Cambio \$0.00

GROUPBOX

BUTTON

DATAGRIDVIEW