Controles de usuario

Apunte Nro 1

PROGRAMACIÓN II

Profesor: DARIO CARDACCI

Controles de usuario

A medida que diseña y modifica la interfaz de usuario de las aplicaciones de Windows Forms, deberá agregar, alinear y colocar los controles de usuario. Los controles son objetos contenidos dentro de objetos de formularios. Cada tipo de control tiene su propio conjunto de propiedades, métodos y eventos que lo hacen adecuado para un fin determinado. Puede manipular los controles del diseñador y escribir código para agregar controles de forma dinámica en el entorno de tiempo de ejecución.

En particular nos concentraremos en los siguientes controles aunque existen muchos más que lo invitamos a que los investigue cuando agote los que están a continuación:

**1 Textbox.**

Los cuadros de texto se utilizan para obtener la entrada del usuario o para mostrar texto. El control se utiliza para texto editable, aunque también se puede convertir en de solo lectura. Los cuadros de texto pueden mostrar varias líneas, ajustar el texto al tamaño del control y agregar formato básico. El TextBox control permite un único formato para el texto que se muestra o se escribe en el control.

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/textbox-control-windows-forms>

**2 Label.**

Se usan para mostrar texto o imágenes que el usuario no puede editar. Se usan para identificar objetos en un formulario, para proporcionar una descripción de lo que hará un determinado control si se hace clic en él, por ejemplo, o para mostrar información en respuesta a un evento o proceso en tiempo de ejecución de la aplicación. El control de [Label](https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.label) no puede recibir el foco, también se puede usar para crear teclas de acceso para otros controles.

<https://docs.microsoft.com/es-es/dotnet/framework/winforms/controls/label-control-windows-forms>

**3 Button.**

Se utiliza para colocar un botón que al oprimirlo produce que se ejecute una acción.<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.button?view=netcore-3.1>

**4 Checkbox.**

Use para CheckBox dar al usuario una opción, como true/false o yes/no. El CheckBox control puede mostrar una imagen, texto o ambos.

CheckBox Los controles y tienen una función similar: permiten RadioButton al usuario elegir entre una lista de opciones. CheckBox Los controles permiten al usuario elegir una combinación de opciones. En cambio, RadioButton los controles permiten a un usuario elegir entre opciones mutuamente excluyentes.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.checkbox?view=windowsdesktop-6.0>

**5 Radiobutton.**

Cuando el usuario selecciona un botón de opción dentro de un grupo, los demás se borran automáticamente. Todos RadioButton los controles de un contenedor determinado, como, constituyen un Form grupo. Para crear varios grupos en un formulario, coloque cada grupo en su propio contenedor, como un GroupBox control o Panel.

Use la Checked propiedad para obtener o establecer el estado de un RadioButton . La apariencia del botón de opción se puede modificar para que aparezca como un botón de estilo de alternancia o como un botón de opción estándar estableciendo la Appearance propiedad

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.radiobutton?view=windowsdesktop-6.0>

**6 GrupBox.**

Muestra GroupBox un marco alrededor de un grupo de controles con o sin un título. Use para GroupBox agrupar lógicamente una colección de controles en un formulario. El cuadro de grupo es un control de contenedor que se puede usar para definir grupos de controles.

El uso típico de un cuadro de grupo es contener un grupo lógico de RadioButton controles. Si tiene dos cuadros de grupo, cada uno de los cuales contiene varios botones de opción, cada grupo de botones es mutuamente excluyente, estableciendo un valor de opción por grupo.

Puede agregar controles a GroupBox mediante el método de la propiedad Add Controls.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.groupbox?view=windowsdesktop-6.0>

**7 Listbox.**

El control permite mostrar una lista de elementos al usuario que el usuario ListBox puede seleccionar haciendo clic en él. Un ListBox control puede proporcionar una o varias selecciones mediante la propiedad SelectionMode. también proporciona la propiedad para habilitar la presentación de elementos en columnas en lugar de ListBox una lista vertical recta de MultiColumn elementos. Con esto, el control puede mostrar elementos más visibles y el usuario ya no necesita desplazarse a un elemento.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.listbox?view=windowsdesktop-6.0>

**8 Combobox.**

Muestra un cuadro de texto combinado con, que permite al usuario ComboBox seleccionar elementos de la lista o escribir un nuevo ListBox valor.

La propiedad especifica si la lista siempre se muestra o si la lista DropDownStyle se muestra en una lista desplegable. La DropDownStyle propiedad también especifica si la parte del texto se puede editar. No hay ninguna opción para mostrar siempre la lista y no permitir la entrada de un nuevo valor.

Para agregar o quitar objetos de la lista en tiempo de ejecución, use métodos de la clase (a través de ComboBox.ObjectCollection Items la propiedad de ComboBox). Puede asignar una matriz de referencias de objeto con el AddRange método. La lista muestra el valor de cadena predeterminado para cada objeto. Puede agregar objetos individuales con el Add método. Puede eliminar elementos con el Remove método o borrar toda la lista con el método Clear.

Además de la funcionalidad de visualización y selección, también proporciona características que permiten agregar eficazmente elementos a y para buscar texto dentro de los elementos ComboBox ComboBox de la lista. Con los métodos y, puede agregar un gran número de elementos a sin que el control se vuelva a dibujar cada vez que se agregue un elemento BeginUpdate EndUpdate a la ComboBox lista. Los FindString FindStringExact métodos y permiten buscar un elemento en la lista que contiene una cadena de búsqueda específica.

https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.combobox?view=windowsdesktop-6.0

**9 Monthcalendar.**

El MonthCalendar control permite al usuario seleccionar una fecha mediante una presentación visual. Puede limitar la fecha y las horas que se pueden seleccionar estableciendo las propiedades MinDate y MaxDate.

El MonthCalendar sistema operativo dibuja el control, por lo que el evento nunca se Paint genera. Si necesita proporcionar una apariencia personalizada al control, debe invalidar el método, llamar a la implementación base de y, a continuación, MonthCalendar OnPrint realizar un dibujo OnPrint personalizado.

Si necesita un formato de fecha personalizado y una selección limitada a una sola fecha, considere la posibilidad de usar DateTimePicker un control en lugar de MonthCalendar . El uso DateTimePicker de elimina gran parte de la necesidad de validar los valores de fecha y hora.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.monthcalendar?view=windowsdesktop-6.0>

**10 ProgressBar.**

Un control consta de una ventana que se rellena, de forma predeterminada de izquierda a derecha, a medida ProgressBar que avanza una operación. El control tiene un intervalo y una posición actual.

ProgressBar invalida los metadatos de la Maximum propiedad y establece su valor predeterminado en 100.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.controls.progressbar?view=windowsdesktop-6.0>

**11 DataGridView.**

El DataGridView control proporciona una tabla personalizable para mostrar datos. La DataGridView clase permite la personalización de celdas, filas, columnas y bordes mediante el uso de propiedades como DefaultCellStyle, ColumnHeadersDefaultCellStyle, CellBorderStyle y GridColor.

Puede usar un DataGridView control para mostrar datos con o sin un origen de datos subyacente. Sin especificar un origen de datos, puede crear columnas y filas que contengan datos y agregarlos directamente a DataGridView mediante las propiedades y Rows Columns. También puede usar la colección Rows para tener acceso a objetos y a la propiedad para leer o escribir valores de celda DataGridViewRow DataGridViewRow.Cells directamente. El Item[] indexador también proporciona acceso directo a las celdas.

Como alternativa a rellenar manualmente el control, puede establecer las propiedades y para enlazar a un origen de datos y rellenarlo automáticamente DataSource DataMember con DataGridView datos.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.datagridview?view=windowsdesktop-6.0>

**12 ColorDialog.**

El miembro heredado ShowDialog se debe invocar para crear este cuadro de diálogo común específico. HookProc se puede invalidar para implementar la funcionalidad específica de enlace del cuadro de diálogo. Use Color para recuperar el color seleccionado por el usuario.

Cuando se crea una instancia de ColorDialog , algunas de las propiedades de lectura y escritura se establecen en valores iniciales.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.colordialog?view=windowsdesktop-6.0>

**13 FontDialog.**

El miembro heredado debe ShowDialog invocarse para crear este cuadro de diálogo común específico. HookProc se puede invalidar para implementar la funcionalidad de enlace de cuadro de diálogo específica.

Al crear una instancia de, algunas de las propiedades de lectura FontDialog y escritura se establecen en valores iniciales.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.fontdialog?view=windowsdesktop-6.0>

**14 Linklabel.**

El LinkLabel control es similar a un control con la Label excepción de que puede mostrar un hipervínculo. Se pueden especificar varios hipervínculos en el texto del control. Cada hipervínculo puede realizar una tarea diferente dentro de una aplicación. Por ejemplo, puede usar un hipervínculo para mostrar un sitio web en Microsoft Internet Explorer o para cargar un archivo de registro asociado a una aplicación.

Cada hipervínculo que se muestra en LinkLabel el control es una instancia de la clase LinkLabel.Link. La LinkLabel.Link clase define la información de visualización, el estado y la ubicación del hipervínculo. Además, la propiedad de la clase permite asociar información, como una LinkData dirección URL que se va a LinkLabel.Link mostrar, con el hipervínculo. Cuando un usuario hace clic en un hipervínculo dentro del control, se genera el evento y el objeto que representa el hipervínculo en el que se hizo clic se pasa como parte del objeto que se pasa como parámetro al controlador de LinkClicked LinkLabel.Link LinkLabelLinkClickedEventArgs eventos. Puede usar este objeto para obtener el objeto asociado al hipervínculo en el que el usuario LinkLabel.Link hizo clic. Todos los hipervínculos contenidos en el LinkLabel control se almacenan en la instancia de clase para el control LinkLabel.LinkCollection.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.linklabel?view=windowsdesktop-6.0>

**15 Checkedlistbox.**

Este control presenta una lista de elementos que el usuario puede navegar mediante el teclado o la barra de desplazamiento del lado derecho del control. El usuario puede colocar una marca de verificación por uno o varios elementos y los elementos activados se pueden navegar con CheckedListBox.CheckedItemCollection y CheckedListBox.CheckedIndexCollection.

Para agregar objetos a la lista en tiempo de ejecución, asigne una matriz de referencias de objeto con el AddRange método. A continuación, la lista muestra el valor de cadena predeterminado para cada objeto. Puede agregar elementos individuales a la lista con el Add método.

El CheckedListBox objeto admite tres estados a través de la CheckState enumeración: y Checked Indeterminate Unchecked . Debe establecer el estado de en el código porque la interfaz de usuario de un no proporciona Indeterminate CheckedListBox un mecanismo para hacerlo.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.checkedlistbox?view=windowsdesktop-6.0>

**16 DateTimePicker**

El control se usa para permitir al usuario seleccionar una fecha y hora y mostrar esa fecha y hora DateTimePicker en el formato especificado.

La Value propiedad contiene la fecha y hora actuales en las que se establece el control. Puede usar la Text propiedad o el miembro adecuado de para obtener el valor de fecha y Value hora. Puede limitar las fechas y horas que se pueden seleccionar estableciendo las propiedades MaxDate y MinDate.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.datetimepicker?view=windowsdesktop-6.0>

**17 ListView**

Un control permite mostrar una lista de elementos con texto de elemento y, opcionalmente, un icono para identificar ListView el tipo de elemento. Por ejemplo, la Windows explorador de archivos es similar en apariencia a un ListView control. Muestra una lista de los archivos y carpetas seleccionados actualmente en el árbol. Cada archivo y carpeta muestra un icono asociado a él para ayudar a identificar el tipo de archivo o carpeta. La ListViewItem clase representa un elemento dentro de un control ListView.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.listview?view=windowsdesktop-6.0>

**18 NotifyIcon.**

Especifica un componente que crea un icono en el área de notificación. Esta clase no puede heredarse.

Los iconos del área de notificación son accesos directos a los procesos que se ejecutan en segundo plano de un equipo, como un programa de protección antivirus o un control de volumen. Estos procesos no vienen con sus propias interfaces de usuario. La NotifyIcon clase proporciona una manera de programar esta funcionalidad. La Icon propiedad define el icono que aparece en el área de notificación. Los menús emergentes de un icono se abordan con la ContextMenu propiedad. La Text propiedad asigna texto de información sobre herramientas. Para que el icono se muestre en el área de notificación, la Visible propiedad debe establecerse en true.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.notifyicon?view=windowsdesktop-6.0>

**19 NumericUpDown.**

Un control contiene un único valor numérico que se puede incrementar o disminuir haciendo clic en los botones arriba o abajo NumericUpDown del control. El usuario también puede escribir un valor, a menos que ReadOnly la propiedad esté establecida en true.

Se puede dar formato a la presentación numérica estableciendo las propiedades DecimalPlaces Hexadecimal, o ThousandsSeparator. Para mostrar valores hexadecimales en el control, establezca la Hexadecimal propiedad en true. Para mostrar un separador de miles en números decimales cuando sea necesario, establezca la ThousandsSeparator propiedad en true. Para especificar el número de dígitos que se muestran después del símbolo decimal, establezca la propiedad en el DecimalPlaces número de posiciones decimales que se mostrarán.

Para especificar el intervalo de valores permitido para el control, establezca las propiedades Maximum y Minimum.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.numericupdown?view=windowsdesktop-6.0>

**20 PictureBox.**

Normalmente se usa para mostrar gráficos de un mapa de PictureBox bits, metarchivo, icono, JPEG, GIF o archivo PNG.

Establezca la Image propiedad en el que desea Image mostrar, ya sea en tiempo de diseño o en tiempo de ejecución. También puede especificar la imagen estableciendo la propiedad y cargando la imagen de forma sincrónica mediante el método o ImageLocation de forma asincrónica mediante el método Load LoadAsync.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.picturebox?view=windowsdesktop-6.0>

**21 RichtextBox.**

Con el RichTextBox control, el usuario puede escribir y editar texto. El control también proporciona características de formato más avanzadas que el TextBox control estándar. El texto se puede asignar directamente al control o se puede cargar desde un formato de texto enriquecido (RTF) o un archivo de texto sin formato. Al texto del control se le puede asignar formato de carácter y párrafo.

El RichTextBox control proporciona métodos que proporcionan funcionalidad para abrir y guardar archivos. El LoadFile método permite cargar un archivo de texto RTF o ASCII existente en el control. También puede cargar datos desde un flujo de datos ya abierto. permite SaveFile guardar un archivo en texto RTF o ASCII. De forma similar al LoadFile método, también puede usar el método para guardar en un flujo de datos SaveFile abierto. El RichTextBox control también proporciona características para buscar cadenas de texto. El método se sobrecarga para buscar tanto cadenas de texto como Find caracteres específicos dentro del texto del control.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.richtextbox?view=windowsdesktop-6.0>

**22 ToolTip.**

La ToolTip clase se usa normalmente para alertar a los usuarios del uso previsto de un control. Por ejemplo, puede especificar texto de información sobre herramientas para un control que acepte un nombre, especificando el formato del nombre que se va TextBox a escribir en el control. Además de proporcionar sugerencias, también puede usar la ToolTip clase para proporcionar información de estado en tiempo de ejecución. Por ejemplo, puede usar la clase para mostrar datos de velocidad de conexión y calidad de línea cuando el usuario mueve el puntero a un control que muestra el estado ToolTip PictureBox de conexión a Internet.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.tooltip?view=windowsdesktop-6.0>

**23 TreeView.**

La Nodes colección contiene todos los objetos TreeNode asignados al control TreeView . Los nodos de árbol de esta colección se conocen como nodos de árbol raíz. Cualquier nodo de árbol que se agrega posteriormente a un nodo de árbol raíz se conoce como nodo secundario. Dado que cada uno de ellos puede contener una colección de otros objetos, es posible que le sea difícil determinar la ubicación en la estructura de árbol al recorrer en TreeNode TreeNode iteración la colección. Puede analizar la cadena mediante el valor de cadena para determinar dónde comienza y termina TreeNode.FullPath PathSeparator una TreeNode etiqueta.

https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.treeview?view=windowsdesktop-6.0

**24 WebBrowser.**

El WebBrowser control permite hospedar páginas web y otros documentos habilitados para el explorador en las aplicaciones Windows Forms. Puede usar el control, por ejemplo, para proporcionar asistencia al usuario basada en HTML integrada o funcionalidades Exploración web WebBrowser en la aplicación. Además, puede usar el control para agregar los controles basados en web existentes a las WebBrowser aplicaciones cliente Windows Forms.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.webbrowser?view=windowsdesktop-6.0>

**25 MenuStrip.**

MenuStrip es el contenedor de nivel superior que reemplaza a MainMenu. También proporciona control de claves y varias características de interfaz de documento (MDI). Funcionalmente, ToolStripDropDownItem y funcionan junto con, aunque se ToolStripMenuItem MenuStrip derivan de ToolStripItem.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.menustrip?view=windowsdesktop-6.0>

**26 FolderBrowseDialog.**

Esta clase proporciona una manera de pedir al usuario que examine, cree y, finalmente, seleccione una carpeta. Use esta clase cuando solo desee permitir que el usuario seleccione carpetas, no archivos. La exploración de las carpetas se realiza a través de un control de árbol. Solo se pueden seleccionar carpetas del sistema de archivos; Las carpetas virtuales no pueden.

Normalmente, después de crear un nuevo, se establece en la ubicación desde la FolderBrowserDialog que se va a empezar a RootFolder examinar. Opcionalmente, puede establecer en SelectedPath una ruta de acceso absoluta de una subcarpeta de que se RootFolder seleccionará inicialmente. Opcionalmente, también puede establecer la Description propiedad para proporcionar instrucciones adicionales al usuario. Por último, llame ShowDialog al método para mostrar el cuadro de diálogo al usuario. Cuando se cierra el cuadro de diálogo y el resultado del cuadro de diálogo es, será una cadena que contenga la ruta de ShowDialog acceso a la carpeta DialogResult.

En mi ejemplo ir a Archivo -> Abrir -> Carpeta

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.folderbrowserdialog?view=windowsdesktop-6.0>

**27 OpenFileDialog.**

Esta clase permite comprobar si existe un archivo y abrirlo. La propiedad determina si aparece una casilla de solo lectura ShowReadOnly en el cuadro de diálogo. La propiedad indica si la casilla de solo lectura ReadOnlyChecked está activada.

La mayor parte de la funcionalidad principal de esta clase se encuentra en la FileDialog clase.

En un sistema operativo de derecha a izquierda, al establecer la propiedad del formulario que contiene en se localizan los botones Nombre de archivo, Abrir y Cancelar del RightToLeft RightToLeft.Yes cuadro de diálogo. Si la propiedad no se establece en RightToLeft.Yes, en su lugar se usa texto en inglés.

Si desea dar al usuario la capacidad de seleccionar una carpeta en lugar de un archivo, use en FolderBrowserDialog su lugar.

Probar Archivo -> Abrir -> Archivo

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.openfiledialog?view=windowsdesktop-6.0>

**28 SaveFileDialog.**

Esta clase puede abrir y sobrescribir un archivo existente o crear un archivo nuevo.

La mayor parte de la funcionalidad de esta clase se encuentra en la FileDialog clase.

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.savefiledialog?view=windowsdesktop-6.0>

**29 PrintDialog.**

Al crear una instancia de PrintDialog , las propiedades de lectura y escritura se establecen en valores iniciales.

Para obtener la configuración de impresora modificada por el usuario con PrintDialog , use la propiedad PrinterSettings .

<https://docs.microsoft.com/es-es/dotnet/api/system.windows.forms.printdialog?view=windowsdesktop-6.0>

Tipos de Datos y Variables

<https://docs.microsoft.com/es-es/dotnet/csharp/tour-of-csharp/types-and-variables>

Arrays

<https://docs.microsoft.com/es-es/dotnet/csharp/programming-guide/arrays/>

Estructuras de decisión

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/if-else>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/switch>

Estructuras de repetición

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/do>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/for>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/foreach-in>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/while>

Instrucciones de salto

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/break>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/continue>

<https://docs.microsoft.com/es-es/dotnet/csharp/language-reference/keywords/return>