PROGRAMACIÓN I

GUÍA DE REVISIÓN

CONCEPTUAL

UNIDAD I

Introducción a la programación visual I

1. **¿Qué característica posee la programación visual?**

Es un lenguaje de programación que utiliza una interfaz visual que nos permite programar en un entorno gráfico. Permite realizar un gran número de tareas sin la necesidad de escribir una gran variedad de código.

1. **¿Qué es un control de usuario?**

Los controles de usuario son contenedores en los que puede colocar controles de formato y de servidor Web

1. **¿Cuáles son los tres elementos constitutivos que un control de usuario debe poseer?**
   1. Evento: acción que se desencadena por una situación en especial
   2. Métodos: Funciones que representan las acciones que van a hacer los elementos
   3. Propiedades: características de los elementos.
2. **¿Para qué se utiliza la ventana denominada Solution Explorer?**

Es una ventana de herramientas que muestra el contenido de una solución, el cual incluye proyectos de la solución y elementos de cada proyecto.

1. **¿Para qué se utiliza la ventana denominada Properties?**

La ventana Properties muestra distintos tipos de campos de edición, según las necesidades de una propiedad determinada. Entre estos campos de edición se incluyen cuadros de edición, listas desplegables y vínculos a cuadros de diálogo de editor personalizado.

1. **¿Para qué se utiliza la ventana denominada Toolbox?**

Muestra los controles de Visual Basic estándar además de los controles ActiveX y los objetos insertables que ha agregado al proyecto.

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Button.**

|  |  |
| --- | --- |
| **Propiedades** | |
| BackColor | Obtiene o establece el color de fondo del control |
| Cursor | Obtiene o establece el cursor que se muestra cuando el puntero del mouse se sitúa sobre el control. |
| DialogResult | Obtiene o establece un valor que se devuelve al formulario principal cuando se hace clic en el botón. |
| Enabled | Obtiene o establece un valor que indica si el control puede responder a la interacción del usuario. |
| FlatStyle | Obtiene o establece la apariencia de estilo plano del control de botón |
| Focused | Obtiene un valor que indica si el control tiene el foco de entrada |
| Font | Obtiene o establece la fuente del texto que muestra el control. |
| Handle | Obtiene el identificador de ventana al que está enlazado el control. |
| Image | Obtiene o establece la imagen que se muestra en un control de botón. |
| Name | Obtiene o establece el nombre del control |
| **Métodos** | |
| BringToFront() | Coloca el control al principio del orden Z |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles. |
| CreateHandle | Crea un identificador para el control. |
| Dispose | Libera todos los recursos que usa Component |
| Focus() | Establece el foco de entrada en el control. |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetType() | Obtiene el “Type” de la instancia actual |
| OnClick(EventArgs) | Genera el evento “Click” |
| PerformClick() | Genera un Click evento para un botón |
| Refresh() | Obliga al control a invalidar su área cliente y, acto seguido, obliga a que vuelva a dibujarse el control y sus controles secundarios |
| **Eventos de Control** | |
| Click | Se produce cuando se hace clic en el control. |
| Enter | Se produce cuando se entra en el control |
| FontChange | Se produce cuando cambia el valor de la propiedad Font |
| Leave | Se produce cuando el foco de entrada deja el control. |
| LostFocus | Se produce cuando el control pierde el foco |
| MouseCaptureChanged | Aparece cuando el control pierde la captura del mouse. |
| MouseClick | Se produce cuando se hace clic con el mouse en el control |
| Paint | Se produce cuando vuelve a dibujarse el control |
| Resized | Se produce cuando se cambia el tamaño del control. |
| TextChanged | Se produce cuando cambia el valor de la propiedad Text |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control CheckBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Bottom | Obtiene la distancia, en puntos, entre el borde inferior del control CheckBox y el borde superior de la hoja de cálculo. |
| Creator | Obtiene un valor que indica la aplicación en la que se creó el control Check Box. |
| Index | Obtiene el número de índice del control CheckBox dentro de la colección de OLEObjects de la hoja |
| Left | Obtiene o establece la distancia, expresada en puntos, entre el borde izquierdo del control Check Box y el borde izquierdo de la hoja de cálculo. |
| Locked | Obtiene o establece un valor que indica si se puede modificar el control CheckBox en tiempo de diseño cuando la hoja está protegida |
| OLEType | Obtiene un valor que representa el tipo de objeto OLE |
| Placement | Obtiene o establece un objeto que indica la manera en que el control CheckBox se adjunta a las celdas que están debajo de é |
| SourceName | Obtiene o establece el nombre de origen del vínculo del control CheckBox |
| Visible | Obtiene o establece un valor que representa si está visible el control CheckBox |
| Width | Establece o devuelve el ancho de CheckBox en puntos |
| ZOrder | Obtiene la posición en el orden Z del control CheckBox |
| **Métodos** | |
| Activate() | Activa el CheckBox |
| CopyPicture(XlPictureAppearance, XlCopyPictureFormat) | Copia el CheckBox al Portapapeles como una imagen. |
| Delete() | Elimina de la hoja de cálculo un control CheckBox creado dinámicamente y lo quita de la colección ControlCollection. |
| FindForm() | Recupera el formulario en el que se encuentra el control |
| Focus() | Establece el foco de entrada en el control |
| GetAutoSizemode() | Recupera un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container |
| SetVisibleCore(Boolean) | Establece el CheckBox en el estado de visibilidad especificado. |
| Update() | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. |
| **Eventos de Control** | |
| DataBinding | Se produce cuando el control de servidor se enlaza a un origen de datos |
| CheckedChanged | Se produce cuando el valor de la propiedad Checked cambia de un envío a otro envío en el servidor |
| Init | Tiene lugar al inicializar el control de servidor, que es el primer paso en su ciclo de vida. |
| Disposed | Se produce cuando un control de servidor se libera de la memoria, lo que constituye la última fase del período de duración de un control de servidor cuando se solicita una página ASP.NET |
| Load | Se produce cuando el control de servidor se carga en el objeto Page |
| PreRender | Se produce una vez que se carga el objeto Control, pero antes de su representación |
| GotFocus | Se produce cuando este elemento recibe el foco lógico |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control CheckedListBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AllowSelection | Obtiene un valor que indica si la ListBox habilita actualmente la selección de elementos de lista |
| BorderStyle | Obtiene o establece el tipo de borde que se dibuja alrededor del ListBox |
| CanEnableIme | Obtiene un valor que indica si la propiedad ImeMode puede establecerse en un valor activo para habilitar la compatibilidad con el IME |
| CanFocus | Obtiene un valor que indica si el control puede recibir el foco |
| CanRaiseEvents | Determina si se pueden producir eventos en el control |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| CheckOnClick | Obtiene o establece un valor que indica si la casilla debería alternar al seleccionar un elemento. |
| CompanyName | Obtiene el nombre de la compañía o del creador de la aplicación que contiene el control. |
| Cursor | Obtiene o establece el cursor que se muestra cuando el puntero del mouse se sitúa sobre el control |
| ForeColor | Obtiene o establece el color de primer plano del control |
| **Métodos** | |
| ClearSelected() | Anula la selección de todos los elementos de la ListBox |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles |
| CreateControlsInstance() | Crea una nueva instancia de la colección de controles para el control. |
| CreateHandle() | Crea un identificador para el control |
| DestroyHandle() | Destruye el identificador asociado a este control |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| FindForm() | Recupera el formulario en el que se encuentra el control. |
| GetHashCode() | Sirve como la función hash predeterminada |
| GetPreferredSize(Size) | Recupera el tamaño de un área rectangular en la que se puede encajar un control. |
| Hide() | Oculta el control al usuario |
| **Eventos del Control** | |
| BindingContextChanged | Se produce cuando cambia el valor de la propiedad BindingContext |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado |
| ContextMenuChanged | Se produce cuando cambia el valor de la propiedad ContextMenu |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| CursorChanged | Se produce cuando cambia el valor de la propiedad Cursor |
| DockChanged | Se produce cuando cambia el valor de la propiedad Dock |
| DragDrop | Se produce cuando se completa una operación de arrastrar y colocar |
| Enter | Se produce cuando se entra en el control |
| Format | Se produce cuando el control se enlaza a un valor de datos |
| GiveFeedback | Se produce durante una operación de arrastre |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Combobox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleRole | Obtiene o establece el rol accesible del control |
| Anchor | 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 |
| AutoCompleteMode | Obtiene o establece una opción que controla cómo funciona la finalización automática para ComboBox |
| AutoScrollOffSet | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView |
| Bounds | Obtiene o establece el tamaño y la ubicación del control incluyendo sus elementos no cliente, en píxeles, con respecto al control primario |
| CanRaiseEvents | Determina si se pueden producir eventos en el control |
| ClientRectangle | Obtiene el rectángulo que representa el área cliente del control |
| ClientSize | Obtiene o establece el alto y el ancho del área cliente del control |
| Controls | Obtiene la colección de controles que contiene el control |
| DataBindings | Obtiene los enlaces de datos del control |
| **Métodos** | |
| CreateControlInstance() | Crea una nueva instancia de la colección de controles para el control. |
| Dispose() | Libera todos los recursos que usa Component |
| DoDragDrop(Object, DragDropEffects) | Inicia una operación de arrastrar y colocar. |
| EndUpdate() | Reanuda el dibujo del control ComboBox después de que el método BeginUpdate suspenda el dibujo |
| Finalize() | Libera recursos no administrados y realiza otras operaciones de limpieza antes de que se reclame el elemento Component durante la recolección de elementos no usados. |
| GetLifeTimeService() | Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia |
| Invoke(Delegate) | Ejecuta el delegado especificado en el subproceso que posee el identificador de ventana subyacente del control |
| IsInputChar(Char) | Determina si un carácter es un carácter de entrada que el control reconoce |
| MemberwsieClone(Boolean) | Crea una copia superficial del elemento actual MarshalByRefObject objeto. |
| OnFormatInfoChanged(EventArgs) | Genera el evento FormatInfoChanged |
| **Eventos de Control** | |
| CausesValidationChanged | Se produce cuando cambia el valor de la propiedad CausesValidation |
| ContextMenuStripChanged | Se produce cuando cambia el valor de la propiedad ContextMenuStrip |
| DisplayMemberChanged | Se produce cuando cambia la propiedad DisplayMember |
| DockChanged | Se produce cuando cambia el valor de la propiedad Dock |
| DragDrop | Se produce cuando se completa una operación de arrastrar y colocar |
| DragOver | Se produce cuando se arrastra un objeto sobre los límites del control |
| DrawItem | Se produce cuando cambia la apariencia de un control ComboBox dibujado por el propietario |
| Format | Se produce cuando el control se enlaza a un valor de datos. |
| ImeModeChanged | Se produce cuando ha cambiado la propiedad ImeMode |
| Layout | Se produce cuando un control debe volver a colocar sus controles secundarios. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control DateTimePicker.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Anchor | 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 |
| Bottom | Obtiene la distancia, en píxeles, que existe entre el borde inferior del control y el borde superior del área cliente de su contenedor. |
| Bounds | Obtiene o establece el tamaño y la ubicación del control incluyendo sus elementos no cliente, en píxeles, con respecto al control primario |
| CalendarFont | Obtiene o establece el estilo de fuente que se aplica al calendario |
| CalendarTitleBackColor | Obtiene o establece el color de fondo del título del calendario |
| CanFocus | Obtiene un valor que indica si el control puede recibir el foco. |
| CanSelect | Obtiene un valor que indica si el control se puede seleccionar |
| ClientRectangle | Obtiene el rectángulo que representa el área cliente del control. |
| Controls | Obtiene la colección de controles que contiene el control. |
| CreateParams | Obtiene los parámetros de creación necesarios cuando se crea el identificador del control |
| **Métodos** | |
| DefWndProc(Message) | Envía el mensaje especificado al procedimiento de ventana predeterminado. |
| Finalize() | Libera recursos no administrados y realiza otras operaciones de limpieza antes de que se reclame el elemento Component durante la recolección de elementos no usados |
| GetTopLevel() | Determina si el control es de nivel superior |
| IniLayout() | Se llama a este método cuando el control se ha agregado a otro contenedor. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| OnPaintBackgrounds(PaintEventArgs) | Pinta el fondo del control. |
| PerformLayout() | Obliga al control a aplicar la lógica de diseño a todos sus controles secundarios |
| PointToClient(Point) | Calcula la ubicación del punto especificado de la pantalla, en coordenadas de cliente |
| Refresh() | Obliga al control a invalidar su área cliente y, acto seguido, obliga a que vuelva a dibujarse el control y sus controles secundarios |
| ResetCursor() | Restablece el valor predeterminado de la propiedad Cursor |
| **Eventos de Control** | |
| CloseUp | Se produce cuando se descarta el calendario desplegable y desaparece. |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| DpiChangedBeforeParent | Se produce al cambiar la configuración de PPP para un control mediante programación antes de tener lugar un evento de cambio de PPP para su control o formulario principal. |
| FormatChanged | Se produce cuando cambia el valor de la propiedad Format |
| HelpRequested | Se produce cuando el usuario solicita ayuda para un control |
| LocationChanged | Se produce cuando cambia el valor de la propiedad Location |
| PaddingChanged | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código.Se produce cuando cambia el valor de la propiedad Padding |
| RegionChanged | Se produce cuando cambia el valor de la propiedad Region |
| TabStopChanged | Se produce cuando cambia el valor de la propiedad TabStop |
| Validating | Se produce cuando el control se está validando |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Label.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AllowDrop | Obtiene o establece un valor que indica si el control puede aceptar los datos que el usuario arrastra al mismo. |
| AutoEllipsis | Obtiene o establece un valor que indica si los puntos suspensivos (...) de caracteres aparece en el borde derecho de la Label, que indica que el Label texto se extiende más allá de la longitud especificada de la Label |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView |
| CanEnableIme | Obtiene un valor que indica si la propiedad ImeMode puede establecerse en un valor activo para habilitar la compatibilidad con el IME. |
| Created | Obtiene un valor que indica si se ha creado el control. |
| DeviceDpi | Obtiene el valor de PPP de la pantalla donde se muestra actualmente el control. |
| DoubleBuffered | 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. |
| FlatsStyle | Obtiene o establece la apariencia de estilo plano del control de etiqueta. |
| HasChildren | Obtiene un valor que indica si el control contiene uno o más controles secundarios |
| ImeModeBase | Obtiene o establece el modo IME de un control. |
| **Métodos** | |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control. |
| CreateObjRef(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar a un proxy que se utiliza para comunicarse con un objeto remoto. |
| DrawToBitmap(Bitmap, Rectangle) | Permite la representación del mapa de bits especificado |
| GetNextControl(Control, Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios |
| InitializeLifetimeService() | Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia |
| IsInputChar(Char) | Determina si un carácter es un carácter de entrada que el control reconoce |
| IsInputKeys(Keys) | Determina si la tecla especificada es una tecla de entrada normal o una tecla especial que requiere preprocesamiento. |
| OnNotifyMessage(Message) | Notifica al control los mensajes de Windows |
| OnParentEnabled(EventArgs) | Provoca el evento EnabledChanged cuando cambia el valor de la propiedad Enabled del contenedor del control |
| PreProcessControlMessage  (Message) | Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos |
| **Eventos de Control** | |
| CausesValitationChanged | Se produce cuando cambia el valor de la propiedad CausesValidation |
| HelpRequested | Se produce cuando el usuario solicita ayuda para un control. |
| MarginChanged | Se produce cuando cambia el margen del control |
| QueryAccessibilityHelp | Se produce cuando AccessibleObject proporciona ayuda para aplicaciones de accesibilidad. |
| SystemColorsChanged | Se produce cuando se modifican los colores del sistema |
| TextAlingChanged | Se produce cuando cambia el valor de la propiedad TextAlign |
| QueryContinueDrag | 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 |
| MouseDown | Se produce cuando el puntero del mouse se encuentra sobre el control y se presiona un botón del mouse |
| Paint | Se produce cuando vuelve a dibujarse el control |
| TabIndexChanged | Se produce cuando cambia el valor de la propiedad TabIndex |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control LinkLabel.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan |
| ActiveLinkColor | Obtiene o establece el color utilizado para mostrar un vínculo activo |
| DisabledLinkColor | Obtiene o establece el color utilizado para mostrar un vínculo deshabilitado |
| Disposing | Obtiene un valor que indica si la clase base Control está en el proceso de eliminación |
| ImageAling | Obtiene o establece la alineación de una imagen mostrada en el control |
| ImageList | Obtiene o establece el ImageList que contiene las imágenes para mostrar en el Label control |
| IsAccessible | Obtiene o establece un valor que indica si el control es visible para las aplicaciones de accesibilidad. |
| IsDispose | Obtiene un valor que indica si el control se ha eliminado |
| IsMirrored | Obtiene un valor que indica si el control está reflejado |
| **Métodos** | |
| InitLayout() | Se llama a este método cuando el control se ha agregado a otro contenedor. |
| LogicalToDeviceUnits(Int32) | Convierte un valor de PPP lógico en su valor de PPP DeviceUnit equivalente |
| NotifyInvalidate(Rectangle) | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. Genera el evento Invalidated con una región especificada del control para invalidar |
| OnCreateControl() | Provoca el método CreateControl |
| OnLinkCliked(LinkLabelLinkClikedEventArgs) | Genera el evento LinkClicked |
| OnMouseLeave(EventArgs) | Genera el evento OnMouseLeave |
| OnPrint(PaintEventArgs) | Genera el evento Paint |
| ProcessCmdKeys(Message, Keys) | Procesa una tecla de comando. |
| ProcessMnemonic(Char) | Procesa un carácter de tecla de acceso |
| Refresh() | Obliga al control a invalidar su área cliente y, acto seguido, obliga a que vuelva a dibujarse el control y sus controles secundarios |
| **Eventos de Control** | |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| ControlRemoved | Se produce cuando se quita un control de Control.ControlCollection |
| LinkClicked | Se produce cuando se hace clic en un vínculo del control |
| MouseCaptureChanged | Aparece cuando el control pierde la captura del mouse. |
| MouseDoubleClick | Se produce cuando se hace doble clic con el mouse en el control |
| DragOver | Se produce cuando se arrastra un objeto sobre los límites del control |
| DragEnter | Se produce cuando se arrastra un objeto dentro de los límites del control. |
| Disposed | Se produce cuando el componente se elimina mediante una llamada a la Dispose (método) |
| BackgroundImageLayoutChanged | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. Se produce cuando cambia la propiedad BackgroungImageChanged |
| KeyUp | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. Se produce cuando el usuario suelta una tecla mientras la etiqueta tiene el foco |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ListBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| ColumnWidth | Obtiene o establece el ancho de las columnas en un control ListBox |
| CompanyName | Obtiene el nombre de la compañía o del creador de la aplicación que contiene el control. |
| ContainsFocus | Obtiene un valor que indica si el control, o uno de sus controles secundarios, tiene el foco de entrada en la actualidad |
| ContextMenu | Obtiene o establece el menú contextual asociado al control. |
| DataManager | Obtiene el CurrencyManager asociado a este control |
| DataSource | Obtiene o establece el origen de datos de este objeto ListControl |
| DesingMode | Obtiene un valor que indica si la Component está actualmente en modo de diseño |
| DisplayMember | Obtiene o establece la propiedad que se va a mostrar para este ListControl |
| Events | Obtiene la lista de controladores de eventos que se adjuntará a este Component |
| FormatInfo | Obtiene o establece la IFormatProvider que proporciona el comportamiento de formato personalizado. |
| **Métodos** | |
| BeginInvoke(Delegate) | Ejecuta el delegado especificado de forma asincrónica en el subproceso donde se creó el identificador subyacente del control. |
| ClearSelected() | Anula la selección de todos los elementos de la ListBox |
| CreateItemCollection() | Crea una nueva instancia de la colección de elementos |
| GetItemText(Object) | Devuelve la representación de texto del elemento especificado |
| GetItemHeight(Int32) | Devuelve el alto de un elemento del control ListBox |
| GetItemRectangle(Int32) | Devuelve el rectángulo delimitador de un elemento en el ListBox |
| GetPreferredSize(Size) | Recupera el tamaño de un área rectangular en la que se puede encajar un control. |
| GetTopLevel() | Determina si el control es de nivel superior. |
| ProcessDialogChar(Char) | Procesa un carácter de diálogo |
| ProcessDialogKey(Key) | Procesa una tecla de diálogo |
| **Eventos de Control** | |
| DrawItem | Se produce cuando cambia la apariencia de un control ListBox dibujado por el propietario |
| Format | Se produce cuando el control se enlaza a un valor de datos |
| FormatInfoChanged | Se produce cuando cambia el valor de la propiedad FormatInfo |
| FormatStringChanged | Se produce cuando el valor de la FormatString cambios de propiedad |
| FormattingEnabledChanged | Se produce cuando cambia el valor de la propiedad FormattingEnabled |
| MeasureItem | Se produce cuando un dibujado por el propietario ListBox se crea y se determina el tamaño de los elementos de lista |
| PreviewKeyDown | Se produce antes que el evento KeyDown cuando se presiona una tecla mientras el foco está en este control |
| SelectedIndexChanged | Se produce cuando el SelectedIndex propiedad o SelectedIndices colección ha cambiado |
| SelectedValueChanged | Se produce cuando cambia la propiedad SelectedValue |
| TabStopChanged | Se produce cuando cambia el valor de la propiedad TabStop |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ListView.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que usan las aplicaciones cliente de accesibilidad. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el rol accesible del control. |
| Alingment | Obtiene o establece la alineación de los elementos del control. |
| AllowColumnReorder | Obtiene o establece un valor que indica si el usuario puede arrastrar encabezados de columna para reordenar las columnas del control. |
| Bounds | Obtiene o establece el tamaño y la ubicación del control incluyendo sus elementos no cliente, en píxeles, con respecto al control primario. |
| CanRaiseEvents | Determina si se pueden producir eventos en el control. |
| CheckBoxes | Obtiene o establece un valor que indica si se va a mostrar una casilla junto al elemento del control. |
| Columns | Obtiene la colección de todos los encabezados de columna que aparecen en el control |
| **Métodos** | |
| ArrangeIcons() | Organiza los elementos del control cuando se muestran como iconos, de acuerdo con el valor de la propiedad Alignment |
| AutoresizeColumns (ColumnHeaderAutoResizeStyle) | Cambia el ancho de las columnas según el estilo de cambio de tamaño |
| Clear() | Quita todos los elementos y columnas del control. |
| CreateAccessibilityInstance() | Crea un nuevo objeto de accesibilidad para el control. |
| CreateHandle() | Crea un identificador para el control. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| FindItemWithText(String) | Encuentra el primer objeto ListViewItem que comience con el valor de texto especificado. |
| GetLifetimeService() | Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia. |
| HIde() | Oculta el control al usuario |
| Invalidate() | Invalida toda la superficie del control y hace que se vuelva a dibujar el control. |
| **Eventos de Control** | |
| AfterLabelEdit | Se produce cuando el usuario edita la etiqueta de un elemento. |
| BeforeLabelEdit | Se produce cuando el usuario empieza a editar la etiqueta de un elemento. |
| ColumnClick | Se produce cuando el usuario hace clic en un encabezado de columna en el control de vista de lista. |
| ColumnReordered | Se produce cuando cambia el orden de los encabezados de columna. |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| CursorChanged | Se produce cuando cambia el valor de la propiedad Cursor |
| ItemActivate | Tiene lugar cuando se activa un elemento |
| ItemCheck | Se produce cuando cambia el estado de activación de un elemento. |
| ItemChecked | Se produce cuando cambia el estado de activación de un elemento |
| Resized | Se produce cuando se cambia el tamaño del control. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control MaskedTextBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AllowDrop | Obtiene o establece un valor que indica si el control puede aceptar los datos que el usuario arrastra al mismo. |
| AllowPrompAsInput | Obtiene o establece un valor que indica si el usuario puede escribir PromptChar como datos válidos. |
| Anchor | 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. |
| AsciiOnly | Obtiene o establece un valor que indica si el control MaskedTextBox acepta caracteres no incluidos en el juego de caracteres ASCII. |
| Bottom | Obtiene la distancia, en píxeles, que existe entre el borde inferior del control y el borde superior del área cliente de su contenedor. |
| CanSelect | Obtiene un valor que indica si el control se puede seleccionar. |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| CompanyName | Obtiene el nombre de la compañía o del creador de la aplicación que contiene el control |
| ContextMenu | Obtiene o establece el menú contextual asociado al control. |
| Culture | Obtiene o establece la información de referencia cultural asociada al cuadro de texto enmascarado |
| **Métodos** | |
| AppendText(String) | Anexa texto al texto actual de un cuadro de texto |
| Copy() | Copia la selección actual del cuadro de texto en el **Portapapeles** |
| Cut() | Mueve la selección actual del cuadro de texto al **Portapapeles**. |
| Deselect(All) | Especifica que el valor de la propiedad SelectionLength es cero para que no se seleccione ningún carácter en el control. |
| GetCharFromPosition(Point) | Recupera el carácter más próximo a la ubicación especificada en el control. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container. |
| GetTopLevel() | Determina si el control es de nivel superior. |
| MemberwiseClone() | Crea una copia superficial del Object actual |
| OnNotifyMessage(Message) | Notifica al control los mensajes de Windows |
| Paste() | Reemplaza la selección actual del cuadro de texto con el contenido del **Portapapeles**. |
| **Eventos de Control** | |
| IsOverwriteModeChanged | Se produce después de que el modo de inserción ha cambiado. |
| MaskChanged | Se produce después de que cambie la máscara de entrada |
| MaskInputRejected | Se produce cuando la entrada del usuario o el carácter asignado no coincide con el elemento de formato correspondiente de la máscara de entrada. |
| QueryContinuedDrag | 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. |
| TextAlingChanged | 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. |
| TextChanged | Se produce cuando cambia el valor de la propiedad Text |
| TypeValidationCompleted | Se produce cuando MaskedTextBox ha terminado de analizar el valor actual utilizando la propiedad ValidatingType |
| Validated | Se produce cuando finaliza la validación del control. |
| Validating | Se produce cuando el control se está validando. |
| VisibleChanged | Se produce cuando cambia el valor de la propiedad Visible |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Checkbox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Bottom | Obtiene la distancia, en puntos, entre el borde inferior del control CheckBox y el borde superior de la hoja de cálculo. |
| Creator | Obtiene un valor que indica la aplicación en la que se creó el control Check Box. |
| Index | Obtiene el número de índice del control CheckBox dentro de la colección de OLEObjects de la hoja |
| Left | Obtiene o establece la distancia, expresada en puntos, entre el borde izquierdo del control Check Box y el borde izquierdo de la hoja de cálculo. |
| Locked | Obtiene o establece un valor que indica si se puede modificar el control CheckBox en tiempo de diseño cuando la hoja está protegida |
| OLEType | Obtiene un valor que representa el tipo de objeto OLE |
| Placement | Obtiene o establece un objeto que indica la manera en que el control CheckBox se adjunta a las celdas que están debajo de é |
| SourceName | Obtiene o establece el nombre de origen del vínculo del control CheckBox |
| Visible | Obtiene o establece un valor que representa si está visible el control CheckBox |
| Width | Establece o devuelve el ancho de CheckBox en puntos |
| ZOrder | Obtiene la posición en el orden Z del control CheckBox |
| **Métodos** | |
| Activate() | Activa el CheckBox |
| CopyPicture(XlPictureAppearance, XlCopyPictureFormat) | Copia el CheckBox al Portapapeles como una imagen. |
| Delete() | Elimina de la hoja de cálculo un control CheckBox creado dinámicamente y lo quita de la colección ControlCollection. |
| FindForm() | Recupera el formulario en el que se encuentra el control |
| Focus() | Establece el foco de entrada en el control |
| GetAutoSizemode() | Recupera un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container |
| SetVisibleCore(Boolean) | Establece el CheckBox en el estado de visibilidad especificado. |
| Update() | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. |
| **Eventos de Control** | |
| DataBinding | Se produce cuando el control de servidor se enlaza a un origen de datos |
| CheckedChanged | Se produce cuando el valor de la propiedad Checked cambia de un envío a otro envío en el servidor |
| Init | Tiene lugar al inicializar el control de servidor, que es el primer paso en su ciclo de vida. |
| Disposed | Se produce cuando un control de servidor se libera de la memoria, lo que constituye la última fase del período de duración de un control de servidor cuando se solicita una página ASP.NET |
| Load | Se produce cuando el control de servidor se carga en el objeto Page |
| PreRender | Se produce una vez que se carga el objeto Control, pero antes de su representación |
| GotFocus | Se produce cuando este elemento recibe el foco lógico |
| Unload | Se produce cuando el control de servidor se descarga de la memoria |
| MouseClick | Ocurre cuando el control es seleccionado con el mouse |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control MonthCalendar.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AnnuallyBoldedDates | Obtiene o establece la matriz de objetos DateTime que determina los días del año que se muestran en negrita. |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView(Control) |
| BoldedDates | Obtiene o establece la matriz de objetos DateTime que determina las fechas no periódicas que se muestran en negrita |
| CalendarDimensions | Obtiene o establece el número de columnas y filas de los meses que se muestran. |
| CanSelect | Obtiene un valor que indica si el control se puede seleccionar |
| CausesValidation | 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. |
| DoubleBuffered | Obtiene o establece un valor que indica si el control debe volver a dibujar su superficie mediante un búfer secundario. |
| FirstDayOfWeek | Obtiene o establece el primer día de la semana, tal como aparece en el calendario mensual. |
| LayoutEngine | Obtiene una instancia almacenada en memoria caché del motor de diseño del control. |
| **Métodos** | |
| AddAnnuallyBoldedDate  (DateTime) | Agrega un día que aparece en negrita según una base anual en el calendario mensual. |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles |
| GetDisplayRange(Boolean) | Recupera información de fecha que representa los límites inferior y superior de las fechas mostradas del control. |
| GetNextControl  (Control, Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios. |
| Invoke(Delegate) | Ejecuta el delegado especificado en el subproceso que posee el identificador de ventana subyacente del control. |
| OnDateSelected  (DateRangeEvents) | Genera el evento DateSelected |
| PerformLayout() | Obliga al control a aplicar la lógica de diseño a todos sus controles secundarios |
| PointToClient(Point) | Calcula la ubicación del punto especificado de la pantalla, en coordenadas de cliente. |
| PreProcessMessage(Message) | Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos |
| **Eventos de Control** | |
| DateChanged | Se produce cuando se cambia la fecha seleccionada en el elemento MonthCalendar |
| DateSelected | Se produce cuando el usuario selecciona una fecha de forma explícita mediante el mouse |
| FontChanged | Se produce cuando cambia el valor de la propiedad Font |
| ForeColorChanged | Se produce cuando cambia el valor de la propiedad ForeColor |
| LocationChanged | Se produce cuando cambia el valor de la propiedad Location |
| MouseCaptureChanged | Aparece cuando el control pierde la captura del mouse |
| MouseDoubleClick | Se produce cuando el usuario hace doble clic en el control MonthCalendar con el mouse |
| SystemColorChanged | Se produce cuando se modifican los colores del sistema |
| TabStopChanged | Se produce cuando cambia el valor de la propiedad TabStop |
| TabIndexChanged | Se produce cuando cambia el valor de la propiedad TabIndex |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control NotifyIcon.**

|  |  |
| --- | --- |
| **Propiedades** | |
| BalloonTipIcon | Obtiene o establece el icono que se muestra en el globo de sugerencias asociado al NotifyIcon |
| BalloonTipText | Obtiene o establece el texto que se muestra en el globo de sugerencias asociado al NotifyIcon |
| BalloonTipTitle | Obtiene o establece el título del globo de sugerencias mostrado en el NotifyIcon |
| DesignMode | Obtiene un valor que indica si Component está actualmente en modo de diseño. |
| Events | Obtiene la lista de controladores de eventos asociados a Component |
| Icon | Obtiene o establece el sitio actual |
| Site | Obtiene o establece el ISite de Component |
| Tag | Obtiene o establece un objeto que contiene datos sobre el control NotifyIcon |
| ContextMenu | Obtiene o establece el menú contextual para el icono |
| ContextMenuStrip | Obtiene o establece el menú contextual asociado al NotifyIcon |
| **Métodos** | |
| CreateRefObj(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose() | Libera todos los recursos que usa Component |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual |
| GetHashCode() | Sirve como la función hash predeterminada |
| ShowBalloonTip(Int32) | Muestra un globo de sugerencias en la barra de tareas para el período de tiempo especificado |
| ShowBallonTip(Int32, String, String, ToolTipIcon) | Muestra un globo de sugerencias con el título, el texto y el icono especificado en la barra de tareas para el período de tiempo especificado |
| ToString() | Devuelve una String que contiene el nombre del Component, si existe. Este método no se debe invalidar |
| MemberwiseClone() | Crea una copia superficial del Object actual |
| MemberwiseClone(Boolean) | Crea una copia superficial del objeto MarshalByRefObject actual |
| GetType() | Obtiene el Type de la instancia actual. |
| **Eventos de Control** | |
| BalloonTipClicked | Se produce al hacer clic en el globo de sugerencias |
| BalloonTipClosed | Se produce cuando el usuario cierra el globo de sugerencias. |
| BalloonTipShown | Se produce cuando el globo de sugerencias se muestra en la pantalla |
| DoubleClick | Ocurre cuando el usuario hace doble clic en el icono del área de notificación de la barra de tareas |
| MouseDown | Se produce cuando el usuario presiona el botón del mouse mientras el puntero está sobre el icono del área de notificación de la barra de tareas |
| MouseMove | Se produce cuando el usuario mueve el mouse mientras el puntero está sobre el icono del área de notificación de la barra de tareas |
| MouseUp | Se produce cuando el usuario suelta el botón del mouse mientras el puntero está sobre el icono del área de notificación de la barra de tareas |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose() |
| MouseDoubleClick | Aparece cuando el usuario hace doble clic en el control NotifyIcon con el mouse |
| Click | Ocurre cuando el usuario hace clic en el icono del área de notificación |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control NumericUpDown.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Accelerations | Obtiene una colección de objetos de aceleración ordenados para el control NumericUpDown |
| ActiveControl | Obtiene o establece el control activo en el control contenedor. |
| AutoScaleDimensions | Obtiene o establece las dimensiones a las que se diseñó el control |
| AutoScaleFactor | Obtiene el factor de escala entre las dimensiones de la escala automática actual y en tiempo de diseño. |
| AutoScrollMargin | Obtiene o establece el tamaño del margen de desplazamiento automático |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| ChangingText | Obtiene o establece un valor que indica si la propiedad de texto está siendo modificada internamente por la clase principal correspondiente. |
| ContainsFocus | Obtiene un valor que indica si el control, o uno de sus controles secundarios, tiene el foco de entrada en la actualidad. |
| Created | Obtiene un valor que indica si se ha creado el control |
| Cursor | Obtiene o establece el cursor que se muestra cuando el puntero del mouse se sitúa sobre el control |
| **Métodos** | |
| AdjustFormScrollbars(Boolean) | Ajusta las barras de desplazamiento del contenedor en función de las posiciones actuales de los controles y del control que está seleccionado. |
| BeginInit() | Activa la inicialización de un control NumericUpDown que se emplea en un formulario o que utiliza otro componente. La inicialización se produce en tiempo de ejecución |
| CreateAccessibilityInstance() | Crea un nuevo objeto de accesibilidad para el control. |
| CreateControlInstance() | Crea una nueva instancia de la colección de controles para el control |
| DefWndProc(Message) | Envía el mensaje especificado al procedimiento de ventana predeterminado |
| DownButton() | Reduce el valor del cuadro de número (también conocido como control de flechas) |
| EndInit() | Finaliza la inicialización de un control NumericUpDown que se emplea en un formulario o que utiliza otro componente. La inicialización se produce en tiempo de ejecución |
| GetAutoSizeMode() | Recupera un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada |
| GetNextControl(Control,Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios |
| OnParentEnabledChanged(EventArgs) | Provoca el evento EnabledChanged cuando cambia el valor de la propiedad Enabled del contenedor del control. |
| **Eventos de Control** | |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado |
| Click | Se produce cuando se hace clic en el control. |
| ContextMenuStripChanged | Se produce cuando cambia el valor de la propiedad ContextMenuStrip |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| CursorChanged | Se produce cuando cambia el valor de la propiedad Cursor |
| DragDrop | Se produce cuando se completa una operación de arrastrar y colocar |
| DragOver | Se produce cuando se arrastra un objeto sobre los límites del control |
| ForeColorChanged | Se produce cuando cambia el valor de la propiedad ForeColor |
| HandleDestroyed | Se produce cuando el identificador del control está en proceso de eliminación |
| HelpRequested | Se produce cuando el usuario solicita ayuda para un control. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control PictureBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| CanRaiseEvents | Determina si se pueden producir eventos en el control |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| ClientRectangle | Obtiene el rectángulo que representa el área cliente del control |
| ContainsFocus | Obtiene un valor que indica si el control, o uno de sus controles secundarios, tiene el foco de entrada en la actualidad |
| Controls | Obtiene la colección de controles que contiene el control |
| Created | Obtiene un valor que indica si se ha creado el control |
| DataBindings | Obtiene los enlaces de datos del control |
| DefaultMaximumSize | Obtiene la longitud y el alto, en píxeles, que se especifica como el tamaño máximo predeterminado de un control |
| ErrorImage | Obtiene o establece la imagen que se va a mostrar cuando se produzca un error durante el proceso de carga de la imagen o si se cancela la carga de la imagen |
| ImageLocation | Obtiene o establece la ruta de acceso o la dirección URL de la imagen que se va a mostrar en el control PictureBox |
| **Métodos** | |
| CancelAsync() | Cancela una carga de la imagen asincrónica |
| GetAutoSiceMode() | Recupera un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada. |
| GetPreferredSize(Size) | Recupera el tamaño de un área rectangular en la que se puede encajar un control |
| GetTopLevel() | Determina si el control es de nivel superior |
| IsInputChar(Char) | Determina si un carácter es un carácter de entrada que el control reconoce |
| Load() | Muestra la imagen especificada por la propiedad ImageLocation de PictureBox |
| LoadAsync() | Carga la imagen de forma asincrónica |
| OnParentVisibleChanged(EventArgs) | Provoca el evento VisibleChanged cuando cambia el valor de la propiedad Visible del contenedor del control |
| PointToScreen(point) | Calcula la ubicación del punto especificado de cliente en coordenadas de pantalla |
| PreProcessMessage(Message) | Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos |
| **Eventos de Control** | |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| FontChanged | Se produce cuando cambia el valor de la propiedad Font |
| LoadCompleted | Se produce cuando la operación de carga de imagen asincrónica se ha completado, se ha cancelado o ha provocado una excepción |
| LoadProgressChanged | Se produce cuando ha cambiado el progreso de una operación de carga de imagen asincrónica |
| QueryContinueDrag | 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 |
| SystemColorsChanged | Se produce cuando se modifican los colores del sistema |
| RightToLeftChanged | Se produce cuando cambia el valor de la propiedad RightToLeft |
| StyleChanged | Se produce cuando cambia el estilo del control |
| TabIndexChanged | Se produce cuando cambia el valor de la propiedad TabIndex |
| TextChanged | Se produce cuando cambia el valor de la propiedad Text |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ProgresBar.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Anchor | 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 |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView(Control) |
| BackgroundImageLayout | Obtiene o establece el diseño de la imagen de fondo de la barra de progreso |
| CausesValidation | Obtiene o establece un valor que indica si el control, cuando recibe el foco, hace que se realice una validación de todos los controles que requieren validación cuando reciben el foco |
| ClientSize | Obtiene o establece el alto y el ancho del área cliente del control. |
| DefaultImeMode | Obtiene el modo de Editor de métodos de entrada (IME) predeterminado admitido por el control |
| DesingMode | Obtiene un valor que indica si Component está actualmente en modo de diseño |
| Disposing | Obtiene un valor que indica si la clase base Control está en el proceso de eliminación |
| InvokeRequired | 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. |
| IsHandleCreated | Obtiene un valor que indica si el control tiene un identificador asociado |
| **Métodos** | |
| AccessibilityNotifyClients  (AccessibleEvents, Int32, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado |
| CreateAccessibilityInstance() | Crea un nuevo objeto de accesibilidad para el control |
| DrawToBimap(Bitmap, Rectangle) | Permite la representación del mapa de bits especificado |
| Invalidate() | Invalida toda la superficie del control y hace que se vuelva a dibujar el control |
| Invoke(Delegate, Object[]) | Ejecuta el delegado especificado en el subproceso que posee el identificador de la ventana subyacente del control, con la lista de argumentos especificada. |
| LogicalToDeviceUnits(Int32) | Convierte un valor de PPP lógico en su valor de PPP DeviceUnit equivalente |
| PerfomrStep() | Avanza la posición actual de la barra de progreso en la cantidad especificada por la propiedad Step |
| PreprocessMessage | Preprocesa los mensajes de teclado o de entrada en el bucle de mensajes antes de enviarlos |
| RectangleToScreen(Rectangle) | Calcula el tamaño y la ubicación del rectángulo de cliente especificado, en coordenadas de pantalla |
| Refresh() | Obliga al control a invalidar su área cliente y, acto seguido, obliga a que vuelva a dibujarse el control y sus controles secundarios. |
| **Eventos de Control** | |
| DoubleClick | Se produce cuando el usuario hace doble clic en el control. |
| EnabledChanged | Se produce cuando cambia el valor de la propiedad Enabled |
| HandleCreated | Se produce cuando se crea un identificador para el control |
| HandleCreated | Se produce cuando el identificador del control está en proceso de eliminación |
| MarginChanged | Se produce cuando cambia el margen del control |
| PaddingChanged | Se produce cuando cambia el valor de la propiedad Padding |
| PreviewKeyDown | Se produce antes que el evento KeyDown cuando se presiona una tecla mientras el foco está en este control |
| QueryContinueDrag | 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 |
| StyleChanged | Se produce cuando cambia el estilo del control |
| SystemColorsChanged | Se produce cuando se modifican los colores del sistema |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control RichTextBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AcceptsTab | Obtiene o establece un valor que indica si al presionar la tecla TAB en un control de cuadro de texto multilínea se escribe un carácter TAB en el control en lugar de moverse el foco al siguiente control en el orden de tabulación. |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que usan las aplicaciones cliente de accesibilidad |
| AutoWordSelection | Obtiene o establece un valor que indica si la selección automática de palabras está habilitada |
| BorderStyle | Obtiene o establece el tipo de borde del control de cuadro de texto |
| BulletIndent | Obtiene o establece la sangría que se utiliza en el control RichTextBox cuando se aplica el estilo de viñeta al texto |
| CanRedo | Obtiene un valor que indica si hay acciones que se han producido en RichTextBox que se pueden volver a aplicar |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| ClientSize | Obtiene o establece el alto y el ancho del área cliente del control |
| Controls | Obtiene la colección de controles que contiene el control |
| DefaultCursor | Obtiene o establece el cursor predeterminado para el control. |
| **Métodos** | |
| AppendText(String) | Anexa texto al texto actual de un cuadro de texto |
| CanPaste(DataFormats+Format) | Determina si se puede pegar información desde el Portapapeles en el formato de datos especificado. |
| Clear() | Borra todo el texto del control de cuadro de texto |
| ClearUndo() | Borra la información sobre la última operación realizada del búfer Deshacer del cuadro de texto. |
| DeselectAll() | Especifica que el valor de la propiedad SelectionLength es cero para que no se seleccione ningún carácter en el control |
| GetFirstCharIndexOfCurrentLine() | Recupera el índice del primer carácter de la línea actual. |
| GetNextControl(Control, Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios |
| GetTopLevel() | Determina si el control es de nivel superior |
| LoadFile(String) | Carga un archivo de texto RTF (Formato de texto enriquecido) o ASCII estándar en el control RichTextBox |
| Paste(DataFormats+Format) | Pega el contenido del Portapapeles en el formato de Portapapeles especificado |
| **Eventos de Control** | |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado |
| ContentsResized | Se produce cuando el contenido del control cambia de tamaño |
| ContextMenuStripChanged | Se produce cuando cambia el valor de la propiedad ContextMenuStrip |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose() |
| GiveFeedBack | Se produce durante una operación de arrastre |
| HandleDestroyed | Se produce cuando el identificador del control está en proceso de eliminación |
| HScroll | Se produce cuando el usuario hace clic en la barra de desplazamiento horizontal del control |
| LinkClicked | Se produce cuando el usuario hace clic en un vínculo en el texto del control |
| LocationChanged | Se produce cuando cambia el valor de la propiedad Location |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control TextBox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessKey | Obtiene o establece la clave de acceso que permite navegar rápidamente al control de servidor web. |
| Adapter | Obtiene el adaptador específico del explorador para el control. |
| Attributes | Obtiene la colección de atributos arbitrarios (sólo para su representación) que no corresponden a propiedades del control. |
| AutoCOmpleteType | Obtiene o establece un valor que indica el comportamiento de la función Autocompletar del control |
| BorderWidth | Obtiene o establece el ancho del borde del control de servidor web. |
| CausesValidation | Obtiene o establece un valor que indica si se realiza la validación cuando el control TextBox se establece para que se realice una validación si se produce un postback |
| ClientID | Obtiene el id. de control para marcado HTML que se genera por ASP.NET. |
| CssClass | Obtiene o establece la clase hoja de estilos en cascada (CSS) presentada por el control de servidor web en el cliente. |
| EnabledTheming | Obtiene o establece un valor que indica si los temas se aplican a este control |
| HasAttributes | Obtiene un valor que indica si el control tiene establecido algún atributo |
| **Métodos** | |
| AddParsedSubObject(Object) | Se reemplaza para permitir agregar únicamente controles literales como la propiedad Text |
| ApplyStyle(Style) | Copia en el control Web todos los elementos no vacíos del estilo especificado, sobrescribiendo los elementos de estilo existentes del mismo. Este método lo usan principalmente los programadores de controles. |
| ClearChiledViewState() | Elimina la información sobre el estado de vista de todos los controles secundarios del control de servidor |
| CreateControlCollection() | Crea un nuevo objeto ControlCollection que contendrá los controles secundarios (de literal y servidor) del control de servidor. |
| DataBind() | Enlaza un origen de datos al control de servidor invocado y a todos sus controles secundarios. |
| Dispose() | Habilita un control de servidor para que realice la limpieza final antes de que se libere de la memoria |
| EnsureChildControls() | Determina si el control de servidor contiene controles secundarios. Si no tiene controles secundarios, los crea. |
| IsLiteralContent() | Determina si el control de servidor alberga únicamente contenido literal |
| MapPathSecure(String) | Recupera la ruta de acceso física a la que se asigna una ruta de acceso virtual, absoluta o relativa. |
| RaiseBubbleEvent(Object, EventArgs) | Asigna los orígenes del evento y su información al control principal del control. |
| **Eventos de Control** | |
| DataBinding | Se produce cuando el control de servidor se enlaza a un origen de datos. |
| Init | Tiene lugar al inicializar el control de servidor, que es el primer paso en su ciclo de vida |
| Disposed | Se produce cuando un control de servidor se libera de la memoria, lo que constituye la última fase del período de duración de un control de servidor cuando se solicita una página ASP.NET. |
| Load | Se produce cuando el control de servidor se carga en el objeto Page |
| PreRender | Se produce una vez que se carga el objeto Control, pero antes de su representación |
| TextChanged | Se produce cuando el contenido del cuadro de texto cambia entre cada envío al servidor |
| Unload | Se produce cuando el control de servidor se descarga de la memoria |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control RadioButton.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que usan las aplicaciones cliente de accesibilidad |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan |
| AllowDrop | Obtiene o establece un valor que indica si el control puede aceptar los datos que el usuario arrastra al mismo. |
| Anchor | 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 |
| AutoCheck | Obtiene o establece un valor que indica si el valor de Checked y el aspecto del control cambian automáticamente al hacer clic en el control. |
| AutoEllipsis | Obtiene o establece un valor que indica si el carácter de puntos suspensivos (...) aparece en el borde derecho del control, lo que indica que el texto del control se extiende más allá de la longitud especificada. |
| Bottom | Obtiene la distancia, en píxeles, que existe entre el borde inferior del control y el borde superior del área cliente de su contenedor |
| Bounds | Obtiene o establece el tamaño y la ubicación del control incluyendo sus elementos no cliente, en píxeles, con respecto al control primario. |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| **Métodos** | |
| BringToFront() | Coloca el control al principio del orden Z. |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control |
| CreateAccessibilityInstance() | Crea un nuevo objeto de accesibilidad para el control RadioButton |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles |
| CreateOnjRef(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose(Boolean) | Libera los recursos no administrados que usa ButtonBase y, de forma opcional, libera los recursos administrados. |
| EndInvoke(IAsyncResult) | Recupera el valor devuelto por la operación asincrónica representada por la interfaz IAsyncResult que se pasa |
| FindForm() | Recupera el formulario en el que se encuentra el control. |
| GetNextControl(Control, Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios. |
| GetTopLevel() | Determina si el control es de nivel superior |
| **Eventos de Control** | |
| AppearenceChanged | Se produce cuando cambia el valor de la propiedad Appearance |
| CheckedChanged | Se produce cuando cambia el valor de la propiedad Checked |
| Click | Se produce cuando se hace clic en el control. |
| DockChanged | Se produce cuando cambia el valor de la propiedad Dock |
| DoubleClick | Se produce cuando el usuario hace doble clic en el control RadioButton |
| HandleCreated | Se produce cuando se crea un identificador para el control |
| HandleDestroyed | Se produce cuando el identificador del control está en proceso de eliminación |
| PaddingChanged | Se produce cuando cambia el relleno del control. |
| QueryContinueDrag | 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 |
| Resize | Se produce cuando se cambia el tamaño del control |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Checkbox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Bottom | Obtiene la distancia, en puntos, entre el borde inferior del control CheckBox y el borde superior de la hoja de cálculo. |
| Creator | Obtiene un valor que indica la aplicación en la que se creó el control Check Box. |
| Index | Obtiene el número de índice del control CheckBox dentro de la colección de OLEObjects de la hoja |
| Left | Obtiene o establece la distancia, expresada en puntos, entre el borde izquierdo del control Check Box y el borde izquierdo de la hoja de cálculo. |
| Locked | Obtiene o establece un valor que indica si se puede modificar el control CheckBox en tiempo de diseño cuando la hoja está protegida |
| OLEType | Obtiene un valor que representa el tipo de objeto OLE |
| Placement | Obtiene o establece un objeto que indica la manera en que el control CheckBox se adjunta a las celdas que están debajo de é |
| SourceName | Obtiene o establece el nombre de origen del vínculo del control CheckBox |
| Visible | Obtiene o establece un valor que representa si está visible el control CheckBox |
| Width | Establece o devuelve el ancho de CheckBox en puntos |
| ZOrder | Obtiene la posición en el orden Z del control CheckBox |
| **Métodos** | |
| Activate() | Activa el CheckBox |
| CopyPicture(XlPictureAppearance, XlCopyPictureFormat) | Copia el CheckBox al Portapapeles como una imagen. |
| Delete() | Elimina de la hoja de cálculo un control CheckBox creado dinámicamente y lo quita de la colección ControlCollection. |
| FindForm() | Recupera el formulario en el que se encuentra el control |
| Focus() | Establece el foco de entrada en el control |
| GetAutoSizemode() | Recupera un valor que indica cómo se comportará un control cuando su propiedad AutoSize esté habilitada |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container |
| SetVisibleCore(Boolean) | Establece el CheckBox en el estado de visibilidad especificado. |
| Update() | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. |
| **Eventos de Control** | |
| DataBinding | Se produce cuando el control de servidor se enlaza a un origen de datos |
| CheckedChanged | Se produce cuando el valor de la propiedad Checked cambia de un envío a otro envío en el servidor |
| Init | Tiene lugar al inicializar el control de servidor, que es el primer paso en su ciclo de vida. |
| Disposed | Se produce cuando un control de servidor se libera de la memoria, lo que constituye la última fase del período de duración de un control de servidor cuando se solicita una página ASP.NET |
| Load | Se produce cuando el control de servidor se carga en el objeto Page |
| PreRender | Se produce una vez que se carga el objeto Control, pero antes de su representación |
| GotFocus | Se produce cuando este elemento recibe el foco lógico |
| Unload | Se produce cuando el control de servidor se descarga de la memoria |
| MouseClick | Ocurre cuando el control es seleccionado con el mouse |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ToolTip.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Active | Obtiene o establece un valor que indica si la información sobre herramientas está actualmente activa. |
| AutomaticDelay | Obtiene o establece el retraso automático de la información sobre herramientas |
| AutoPopDelay | Obtiene o establece el período de tiempo que la información sobre herramientas estará visible si el puntero se detiene sobre un control para el que se haya especificado esa información |
| CreateParams | Obtiene los parámetros de creación para la ventana de información sobre herramientas |
| Events | Obtiene la lista de controladores de eventos asociados a Component |
| InitialDelay | Obtiene o establece el tiempo que transcurre hasta que aparece la información sobre herramientas |
| OwnerDraw | Obtiene o establece un valor que indica si la información sobre herramientas la dibuja el sistema operativo o el código que se proporciona. |
| ReshowDelay | Obtiene o establece el período de tiempo que debe transcurrir antes de que aparezca la siguiente ventana de información sobre herramientas cuando el puntero se desplaza de un control a otro |
| ShowAlways | Obtiene o establece un valor que indica si la ventana de información sobre herramientas se muestra aunque el control primario no esté activo. |
| UseAnimation | Obtiene o establece un valor que determina si se debería utilizar un efecto de animación al mostrar la información sobre herramientas. |
| **Métodos** | |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual |
| Finalize() | Libera los recursos no administrados y realiza otras operaciones de limpieza antes de que el recolector de elementos no utilizados reclame Cursor |
| GetHashCode() | Sirve como la función hash predeterminada |
| GetToolTip(Control) | Recupera el texto de información sobre herramientas asociado al control especificado |
| InitializeLifetimeService() | Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia |
| RemoveAll() | Quita todo el texto de información de herramientas asociado al componente de información sobre herramientas |
| SetToolTop(Control, String) | Asocia el texto de información sobre herramientas al control especificado |
| Show(String, IWin32Window, Point, Int32) | Establece el texto de información sobre herramientas asociado al control especificado y, a continuación, muestra la información durante el tiempo especificado en la posición relativa indicada. |
| StopTimer() | Detiene el temporizador que oculta la información sobre herramientas mostrada |
| ToString() | Devuelve una representación de cadena para este control |
| **Eventos de Control** | |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose() |
| Draw | Se produce cuando se dibuja la información sobre herramientas, la propiedad OwnerDraw se establece en true y el valor de la propiedad IsBalloon es false |
| PopUp | Sucede antes de que se muestre inicialmente una información sobre herramientas. Éste es el evento predeterminado para la clase ToolTip |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control TreeView.**

|  |  |
| --- | --- |
| **Propiedades** | |
| ActualHeight | Obtiene el alto representado de este elemento |
| AllowDrop | Obtiene o establece un valor que indica si este elemento se puede usar como destino de una operación de arrastrar y colocar. Es una propiedad de dependencia. |
| AreAnyTouchesCaptured | Obtiene un valor que indica si se captura al menos una entrada táctil en este elemento |
| AreAnyTouchesOver | Obtiene un valor que indica si se presiona al menos una entrada táctil sobre este elemento o sobre cualquier elemento secundario de su árbol visual. |
| BindingGroup | Obtiene o establece el BindingGroup que se usa para el elemento |
| BitmapEffectInput | Obtiene o establece un origen de entrada para el efecto de imagen que se aplica directamente al contenido representado de este elemento. Es una propiedad de dependencia. |
| BorderBrush | Obtiene o establece un pincel que describe el fondo del borde de un control |
| BorderThickness | Obtiene o establece el grosor del borde de un control |
| Clip | Obtiene o establece la geometría usada para definir el contorno del contenido de un elemento. Es una propiedad de dependencia. |
| DataContext | Obtiene o establece el contexto de datos para un elemento cuando participa en un enlace de datos |
| **Métodos** | |
| AddHandler  (ToutedEvent,Delegate) | Agrega un controlador de eventos enrutados para un evento enrutado especificado. Para ello, agrega el controlador a la colección de controladores en el elemento actual. |
| AddLogicalChile(Object) | Agrega el objeto proporcionado al árbol lógico de este elemento |
| ApplyTemplate() | Compila el árbol visual de la plantilla actual si es necesario y devuelve un valor que indica si esta llamada volvió a compilar el árbol visual. |
| ArrangeOverride(Size) | Se llama para organizar y cambiar el tamaño del contenido de un objeto Control |
| BeginStoryboard(Storyboard) | Empieza la secuencia de acciones contenidas en el guion gráfico proporcionado. |
| BringintoView(Rect) | Intenta mostrar el tamaño de la región proporcionado de este elemento, dentro de cualquier región desplazable que lo contenga. |
| CheckAccess() | Determina si el subproceso de la llamada tiene acceso a DispatcherObject |
| ClearValue(DepencyProperty) | Borra el valor local de una propiedad. La propiedad que se va a borrar se especifica mediante un identificador DependencyProperty |
| ExpandSubtree(TreeViewItem) | Expande el control TreeViewItem especificado y todos sus elementos TreeViewItem secundarios |
| FindResource(Object) | Busca un recurso con la clave especificada y produce una excepción si no se encuentra el recurso solicitado. |
| **Eventos de Control** | |
| ContextMenuClosing | Se produce inmediatamente antes de que se cierre cualquier menú contextual en el elemento |
| DragEnter | Se produce cuando el sistema de entrada notifica un evento de arrastre subyacente con este elemento como el destino del arrastre. |
| Drop | Se produce cuando el sistema de entrada notifica un evento de colocación subyacente con este elemento como el destino de la colocación. |
| GotKeyboardFocus | Se produce cuando este elemento recibe el foco del teclado |
| IsHitTestVisibleChanged | Se produce cuando el valor de la propiedad de dependencia IsHitTestVisible cambia en este elemento |
| LayoutUpdated | Se produce cuando cambia el diseño de los diversos elementos visuales asociados al objeto Dispatcher actual. |
| Loaded | Se produce cuando el elemento se presenta, se representa y está listo para la interacción |
| LostStylusCapture | Se produce cuando este elemento pierde la captura del lápiz. |
| ManipulationBoundaryFeedback | Se produce cuando la manipulación encuentra un límite |
| ManipulationInertiaStarting | Se produce cuando el dispositivo de entrada pierde el contacto con el objeto UIElement durante una manipulación y se inicia la inercia |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control WebBrowser.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que usan las aplicaciones cliente de accesibilidad. |
| ActiveXInstance | Obtiene el control ActiveX WebBrowser subyacente. |
| AllowNavegation | Obtiene o establece un valor que indica si el control puede navegar a otra página después de haberse cargado la página inicial. |
| AllowWebBrowserDrop | Obtiene o establece un valor que indica si el control WebBrowser navega hasta los documentos colocados en él |
| Bottom | Obtiene la distancia, en píxeles, que existe entre el borde inferior del control y el borde superior del área cliente de su contenedor |
| CanGoBack | Obtiene un valor que indica si una página anterior del historial de navegación está disponible; de esta forma, el método GoBack() se puede ejecutar correctamente. |
| CanGoForward | Obtiene un valor que indica si una página posterior del historial de navegación está disponible; de esta forma, el método GoForward() se puede ejecutar correctamente |
| ContextMenu | Obtiene o establece el menú contextual asociado al control |
| CreateParams | Obtiene los parámetros de creación necesarios cuando se crea el identificador del control |
| Dock | Obtiene o establece que los bordes del control se acoplarán a su control primario y determina cómo se cambia el tamaño de un control con su elemento primario |
| **Métodos** | |
| AttachInterfaces(Object) | El control llama a este método cuando se ha creado el control ActiveX subyacente |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles |
| CreateSink() | Asocia el control ActiveX subyacente con un cliente que puede controlar eventos de control |
| CreateWebBrowserSiteBase() | Devuelve una referencia al sitio del control ActiveX WebBrowser no administrado, que se puede ampliar para personalizar el control WebBrowser administrado. |
| GetNextControl  (Control,Boolean) | Recupera el siguiente control, hacia delante o hacia atrás, en el orden de tabulación de controles secundarios |
| GoBack() | Hace que el control WebBrowser navegue hasta la página anterior del historial de navegación, si hay alguna disponible. |
| GoForward() | Hace que el control WebBrowser navegue hasta la página siguiente del historial de navegación, si hay alguna disponible |
| GoHome() | Hace que el control WebBrowser navegue hasta la página principal del usuario actual |
| GoSearch() | Hace que el control WebBrowser navegue hasta la página de búsqueda predeterminada del usuario actual. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| **Eventos de Control** | |
| CanGoBackChanged | Se produce cuando cambia el valor de la propiedad CanGoBack |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| DocumentCompleted | Se produce cuando el control WebBrowser termina de cargar un documento |
| DocumentTitleChanged | Se produce cuando cambia el valor de la propiedad DocumentTitle |
| EncryptionLevelChanged | Se produce cuando el control WebBrowser navega hasta un sitio web que usa el cifrado o cuando sale de él |
| FileDownload | Se produce cuando el control WebBrowser descarga un archivo |
| Navigated | Se produce cuando el control WebBrowser navega hasta un nuevo documento y empieza a cargarlo |
| Navigating | Se produce antes de que el control WebBrowser navegue hasta un nuevo documento |
| NewWindow | Se produce antes de que se abra una nueva ventana del explorador |
| ProgressChanged | Se produce cuando el control WebBrowser actualiza la información sobre el progreso de la descarga de un documento al que está navegando |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Grupbox.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el rol accesible del control. |
| AllowDrop | Obtiene o establece un valor que indica si el control permitirá utilizar eventos y operaciones de arrastrar y colocar |
| Capture | Obtiene o establece un valor que indica si el control ha capturado el mouse |
| ClientRectangle | Obtiene el rectángulo que representa el área cliente del control |
| Created | Obtiene un valor que indica si se ha creado el control. |
| DefaultCursor | Obtiene o establece el cursor predeterminado para el control |
| ForeColor | Obtiene o establece el color de primer plano del control |
| LayoutEngine | Obtiene una instancia almacenada en memoria caché del motor de diseño del control |
| **Métodos** | |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control |
| CreateAccessibilityInstance() | Crea un nuevo objeto de accesibilidad para GroupBox |
| DefWndProc(Message) | Envía el mensaje especificado al procedimiento de ventana predeterminad |
| Invoke(Delegate) | Ejecuta el delegado especificado en el subproceso que posee el identificador de ventana subyacente del control. |
| IsInputChar(Char) | Determina si un carácter es un carácter de entrada que el control reconoce |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| MemberwiseClone(Boolean) | Crea una copia superficial del objeto MarshalByRefObject actual. |
| NotifyInvalidate(Rectangle) | Genera el evento Invalidated con una región especificada del control para invalidar |
| OnParentFontChanged(EventArgs) | Provoca el evento FontChanged cuando cambia el valor de la propiedad Font del contenedor del control |
| PerformLayout() | Obliga al control a aplicar la lógica de diseño a todos sus controles secundarios |
| **Eventos de Control** | |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado. |
| ContextMenuStripChanged | Se produce cuando cambia el valor de la propiedad ContextMenuStrip |
| ControlAdded | Se produce cuando se agrega un nuevo control a Control.ControlCollection |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose() |
| HandleCreated | Se produce cuando se crea un identificador para el control |
| MouseCaptureChanged | Aparece cuando el control pierde la captura del mouse |
| Move | Se produce cuando se mueve el control |
| ParentChanged | Se produce cuando cambia el valor de la propiedad Parent |
| QueryAccessibilityHelp | Se produce cuando AccessibleObject proporciona ayuda para aplicaciones de accesibilidad |
| PreviewKeyDown | Se produce antes que el evento KeyDown cuando se presiona una tecla mientras el foco está en este control |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control DataGridView.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AltHTML | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código |
| Automation | Esta API admite la infraestructura producto y no está diseñada para usarse directamente desde el código. |
| Bottom | Obtiene la distancia, en puntos, entre el borde inferior de DataGridView y el borde superior del documento |
| Height | Obtiene o establece el alto del control DataGridView |
| Left | Obtiene o establece la distancia, en puntos, entre el borde izquierdo de DataGridView y el borde izquierdo del documento |
| Right | Obtiene la distancia, en puntos, entre el borde derecho de DataGridView y el borde izquierdo del documento |
| Shape | Obtiene la Shape subyacente de la DataGridView |
| Top | Obtiene o establece la distancia, en puntos, entre el borde superior de DataGridView y el borde superior del documento |
| Visible | Obtiene o establece un valor que representa si está visible el control |
| Width | Establece o devuelve el ancho de DataGridView en puntos |
| **Métodos** | |
| AreAllCellsSelected(Boolean) | Devuelve un valor que indica si todas las celdas del control DataGridView están actualmente seleccionadas |
| AutoResizeColumnHeadersHeight() | Ajusta el alto de los encabezados de columna para adaptarlos al contenido del encabezado de columna más grande |
| AutoResizeColumns() | Ajusta el ancho de todas las columnas para adaptarlo al contenido de todas sus celdas, incluidas las celdas de encabezado |
| AutoRizeRows() | Ajusta el alto de todas las filas para adaptarlo el contenido de todas sus celdas, incluidas las celdas de encabezado |
| CancelEdit() | Cancela el modo de edición para la celda seleccionada actualmente y descarta los cambios. |
| ClearSelection() | Borra la selección actual anulando la selección de todas las celdas seleccionadas |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles |
| DisplayedColumnCount(Boolean) | Devuelve el número de columnas mostradas al usuario |
| EndEdit  (DataGridViewDataErrorContexts) | Confirma y finaliza la operación de edición en la celda actual utilizando el contexto de error especificado. |
| GetCellCount  (DataGridViewElementState) | Obtiene el número de celdas que satisfacen el filtro proporcionado |
| **Eventos de Control** | |
| AllowUserToAddRowsChanged | Se produce cuando cambia el valor de la propiedad AllowUserToAddRows |
| AutoSizeColumnModeChanged | Se produce cuando cambia el valor de la propiedad AutoSizeMode de una columna. |
| BackColorChanged | Se produce cuando cambia el valor de la propiedad BackColor |
| CancelRowEdit | Se produce cuando la propiedad VirtualMode de un control DataGridView es true y se cancela la edición de una fila |
| CellBeginEdit | Tiene lugar cuando el modo de edición comienza por la celda seleccionada |
| CellClick | Se produce cuando se hace clic en cualquier parte de una celda |
| CellErrorTextChanged | Se produce cuando cambia el valor de la propiedad ErrorText de una celda. |
| CellFormatting | Tiene lugar cuando es necesario dar formato al contenido de una celda para mostrarlo |
| CellMouseDown | Tiene lugar cuando el usuario presiona un botón del mouse cuando el puntero se encuentra dentro de los límites de una celda |
| CellParsing | Cuando una celda cuyo valor ha sido modificado deja de estar en modo de edición |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ColorDialog.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AnyColor | Su valor indica si se mostrarán todos los colores o solos los básicos. |
| ShowHelp | Contiene un valor que define sí aparecerá un botón de ayuda o no dentro del DialogBox. |
| SolidColorOnly | Su valor selecciona si el usuario puede ingresar colores sólidos o no en el DialogBox |
| Color | Contiene un color seleccionado por el usuario. |
| Tag | Contiene datos sobre el control. |
| CustomColors | Obtiene un valor custom mostrado en el Dialog Box. |
| Options | Obtiene valores para inicializar el ColorDialog. |
| DesignMode | El valor índice si el DesignMode está activado o no. |
| FullOpen | Obtiene un valor que le permite indicar si los controles para crear un color custom están activados o desactivados. |
| AllowFullOpen | Obtiene o selecciona un valor indicando si el usuario puede usar un DialogBox para definir colores customizados. |
| **Métodos** | |
| CreateObjtRef(Type) | Crea un objeto que contiene toda la información requerida para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose() | Libera todos los recursos utilizados por el componente. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| GetHashCode() | Sirve como la versión default de la función hash. |
| GetType() | Obtiene un tipo de dato. |
| ToString() | Devuelve un string que representa al ColorDialog. |
| RunDialog(IntPtr) | Cuando se invalida en una clase derivada, especifica una DialogBox común. |
| ShowDialog() | Corre una Dialog Box común con un propietario predeterminado- |
| MemeberwiseClone() | Crea una copia superficial del ob¡eto actual. |
| Reset() | Resetea todas las opciones a sus valores predeterminados, el último color seleccionado a negro, y el color custom a sus valores predeterminados. |
| **Eventos de Control** | |
| Disposed | Ocurre cuando el componente es llamado por el método Dispose(). |
| HelpRequest | Ocurre cuando el usuario clickea en el botón de ayuda en un DialogBox. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control FontDialog.**

|  |  |
| --- | --- |
| **Propiedades** | |
| Color | Obtiene o establece el color de la fuente seleccionada. |
| Font | Obtiene o establece la fuente seleccionada. |
| MaxSize | Obtiene o establece el tamaño de punto máximo que un usuario puede seleccionar. |
| MinSize | Obtiene o establece el tamaño de punto mínimo que un usuario puede seleccionar. |
| ShowColor | Obtiene o establece un valor que indica si el cuadro de diálogo muestra la opción de color. |
| ShowApply | Obtiene o establece un valor que indica si el cuadro de diálogo contiene un botón Aplicar. |
| ShowEffects | Obtiene o establece un valor que indica si el cuadro de diálogo contiene controles que permiten al usuario especificar opciones de tachado, subrayado y color del texto. |
| AllowScriptChange | Obtiene o establece un valor que indica si el usuario puede cambiar el juego de caracteres especificado en el cuadro combinado Alfabeto para mostrar un juego de caracteres diferente del que aparece. |
| AllowVectorFonts | Obtiene o establece un valor que indica si se permite la selección de fuentes de vector en el cuadro de diálogo. |
| AllowVerticalFonts | Obtiene o establece un valor que indica si el cuadro de diálogo muestra tanto fuentes horizontales como verticales o solo fuentes horizontales. |
| **Métodos** | |
| CreateObjRef(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose() | Libera todos los recursos que usa Component. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetLifetimeService() | Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container. |
| GetType() | Obtiene el Type de la instancia actual. |
| HookProc(IntPtr, Int32, IntPtr, IntPtr) | Especifica el procedimiento de enlace de cuadros de diálogo comunes que se omite para agregar funcionalidad específica a un cuadro de diálogo común. |
| InitializeLifetimeService() | Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| **Eventos de Control** | |
| Apply | Se produce cuando el usuario hace clic en el botón Aplicar en el cuadro de diálogo de fuentes. |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose(). |
| HelpRequest | Se produce cuando el usuario hace clic en el botón Ayuda de un cuadro de diálogo común. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control OpenFileDialog.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AddExtension | Obtiene o establece un valor que indica si el cuadro de diálogo agrega automáticamente una extensión a un nombre de archivo en caso de que el usuario omita dicha extensión. |
| AutoUpgradeEnabled | Obtiene o establece un valor que indica si esta instancia de FileDialog debe actualizar automáticamente el aspecto y el comportamiento cuando se ejecute en Windows Vista. |
| CanRaiseEvents | Obtiene un valor que indica si el componente puede generar un evento. |
| CheckFileExists | Obtiene o establece un valor que indica si el cuadro de diálogo muestra una advertencia cuando el usuario especifica un nombre de archivo que no existe. |
| CheckPathExists | Obtiene o establece un valor que indica si el cuadro de diálogo muestra una advertencia cuando el usuario especifica una ruta de acceso que no existe. |
| Container | Obtiene IContainer que contiene Component. |
| CustomPlaces | Obtiene la colección de ubicaciones personalizadas de esta instancia de FileDialog. |
| DefaultExt | Obtiene o establece la extensión de nombre de archivo predeterminada. |
| DereferenceLinks | Obtiene o establece un valor que indica si el cuadro de diálogo devuelve la ubicación del archivo al que hace referencia el acceso directo o bien devuelve la ubicación del acceso directo (.lnk). |
| DesignMode | Obtiene un valor que indica si Component está actualmente en modo de diseño. |
| **Métodos** | |
| CreateObjRef(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose() | Libera todos los recursos que usa Component. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetLifetimeService() | Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container. |
| GetType() | Obtiene el Type de la instancia actual. |
| HookProc(IntPtr, Int32, IntPtr, IntPtr) | Define el procedimiento de enlace para cuadros de diálogo comunes que se reemplaza para agregar una funcionalidad específica a un cuadro de diálogo de archivo. |
| InitializeLifetimeService() | Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| **Eventos de Control** | |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose(). |
| FileOk | Se produce cuando el usuario hace clic en el botón Abrir o Guardar de un cuadro de diálogo de archivo. |
| HelpRequest | Se produce cuando el usuario hace clic en el botón Ayuda de un cuadro de diálogo común. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control SaveFileDialog.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AddExtension | Obtiene o establece un valor que indica si el cuadro de diálogo agrega automáticamente una extensión a un nombre de archivo en caso de que el usuario omita dicha extensión. |
| AutoUpgradeEnabled | Obtiene o establece un valor que indica si esta instancia de FileDialog debe actualizar automáticamente el aspecto y el comportamiento cuando se ejecute en Windows Vista. |
| CanRaiseEvents | Obtiene un valor que indica si el componente puede generar un evento. |
| CheckFileExists | Obtiene o establece un valor que indica si el cuadro de diálogo muestra una advertencia cuando el usuario especifica un nombre de archivo que no existe. |
| CheckPathExists | Obtiene o establece un valor que indica si el cuadro de diálogo muestra una advertencia cuando el usuario especifica una ruta de acceso que no existe. |
| Container | Obtiene IContainer que contiene Component. |
| CreatePrompt | Obtiene o establece un valor que indica si el cuadro de diálogo pide permiso al usuario para crear un archivo cuando el usuario especifica un archivo que no existe. |
| CustomPlaces | Obtiene la colección de ubicaciones personalizadas de esta instancia de FileDialog. |
| DefaultExt | Obtiene o establece la extensión de nombre de archivo predeterminada. |
| DereferenceLinks | Obtiene o establece un valor que indica si el cuadro de diálogo devuelve la ubicación del archivo al que hace referencia el acceso directo o bien devuelve la ubicación del acceso directo (.lnk). |
| **Métodos** | |
| CreateObjRef(Type) | Crea un objeto que contiene toda la información relevante necesaria para generar un proxy utilizado para comunicarse con un objeto remoto. |
| Dispose() | Libera todos los recursos que usa Component. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| GetHashCode() | Sirve como la función hash predeterminada. |
| GetLifetimeService() | Recupera el objeto de servicio de duración actual que controla la directiva de duración de esta instancia. |
| GetService(Type) | Devuelve un objeto que representa el servicio suministrado por Component o por Container. |
| GetType() | Obtiene el Type de la instancia actual. |
| HookProc(IntPtr, Int32, IntPtr, IntPtr) | Define el procedimiento de enlace para cuadros de diálogo comunes que se reemplaza para agregar una funcionalidad específica a un cuadro de diálogo de archivo. |
| InitializeLifetimeService() | Obtiene un objeto de servicio de duración para controlar la directiva de duración de esta instancia. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| **Eventos de Control** | |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose(). |
| FileOk | Se produce cuando el usuario hace clic en el botón Abrir o Guardar de un cuadro de diálogo de archivo. |
| HelpRequest | Se produce cuando el usuario hace clic en el botón Ayuda de un cuadro de diálogo común. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ContextMenuStrip.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibilityObject | Obtiene AccessibleObject asignado al control. |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el rol accesible del control. |
| AllowDrop | Obtiene o establece un valor que indica si las operaciones de arrastrar y colocar y la reordenación de elementos se controlan mediante eventos que se implementan. |
| ScaleChildren | Obtiene un valor que determina el escalado de los controles secundarios. |
| AllowMerge | Obtiene o establece un valor que indica si se pueden combinar varios MenuStrip, ToolStripDropDownMenu, ToolStripMenuItem y otros tipos. |
| AllowTransparency | Obtiene o establece un valor que indica si se puede ajustar la propiedad Opacity de opacidad del formulario. |
| AutoClose | Obtiene o establece un valor que indica si el control ToolStripDropDown se debe cerrar automáticamente cuando pierde la activación. |
| **Métodos** | |
| AccessibilityNotifyClients  (AccessibleEvents, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AccessibilityNotifyClients  (AccessibleEvents, Int32, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AdjustFormScrollbars(Boolean) | Ajusta las barras de desplazamiento del contenedor en función de las posiciones actuales de los controles y del control que está seleccionado. |
| BeginInvoke(Delegate) | Ejecuta el delegado especificado de forma asincrónica en el subproceso donde se creó el identificador subyacente del control. |
| BeginInvoke(Delegate, Object[]) | Ejecuta el delegado especificado de forma asincrónica con los argumentos especificados en el subproceso donde se creó el identificador subyacente del control. |
| BringToFront() | Coloca el control al principio del orden Z. |
| Close() | Cierra el control ToolStripDropDown. |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control. |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles. |
| CreateGraphics() | Crea el Graphics para el control. |
| **Eventos de Control** | |
| AutoSizeChanged | Se produce cuando ha cambiado la propiedad AutoSize. |
| BackColorChanged | Se produce cuando cambia el valor de la propiedad BackColor. |
| BackgroundImageChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BackgroundImageLayoutChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BeginDrag | Se produce cuando el usuario empieza a arrastrar el control ToolStrip. |
| BindingContextChanged | Se produce cuando cambia el valor de la propiedad BindingContext. |
| CausesValidationChanged | Se produce cuando cambia la propiedad CausesValidation. |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado. |
| Click | Se produce cuando se hace clic en el control. |
| ClientSizeChanged | Se produce cuando cambia el valor de la propiedad ClientSize. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control MenuStrip.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibilityObject | Obtiene AccessibleObject asignado al control. |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el rol accesible del control. |
| AllowDrop | Obtiene o establece un valor que indica si las operaciones de arrastrar y colocar y la reordenación de elementos se controlan mediante eventos que se implementan. |
| ScaleChildren | Obtiene un valor que determina el escalado de los controles secundarios. |
| AllowMerge | Obtiene o establece un valor que indica si se pueden combinar varios MenuStrip, ToolStripDropDownMenu, ToolStripMenuItem y otros tipos. |
| Anchor | Obtiene o establece los bordes del contenedor al que está enlazado un ToolStrip y determina cómo se cambia el tamaño de un ToolStrip con su elemento primario. |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView(Control). |
| **Métodos** | |
| AccessibilityNotifyClients  (AccessibleEvents, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AccessibilityNotifyClients  (AccessibleEvents, Int32, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AdjustFormScrollbars(Boolean) | Ajusta las barras de desplazamiento del contenedor en función de las posiciones actuales de los controles y del control que está seleccionado. |
| BeginInvoke(Delegate) | Ejecuta el delegado especificado de forma asincrónica en el subproceso donde se creó el identificador subyacente del control. |
| BeginInvoke(Delegate, Object[]) | Ejecuta el delegado especificado de forma asincrónica con los argumentos especificados en el subproceso donde se creó el identificador subyacente del control. |
| BringToFront() | Coloca el control al principio del orden Z. |
| Dispose() | Libera todos los recursos que usa Component. |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control. |
| CreateControl() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles. |
| CreateGraphics() | Crea el Graphics para el control. |
| **Eventos de Control** | |
| AutoSizeChanged | Se produce cuando ha cambiado la propiedad AutoSize. |
| BackColorChanged | Se produce cuando cambia el valor de la propiedad BackColor. |
| BackgroundImageChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BackgroundImageLayoutChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BeginDrag | Se produce cuando el usuario empieza a arrastrar el control ToolStrip. |
| BindingContextChanged | Se produce cuando cambia el valor de la propiedad BindingContext. |
| CausesValidationChanged | Se produce cuando cambia la propiedad CausesValidation. |
| ChangeUICues | Cuando cambian las guías de la interfaz de usuario para el foco o el teclado. |
| Click | Se produce cuando se hace clic en el control. |
| ClientSizeChanged | Se produce cuando cambia el valor de la propiedad ClientSize. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control StatusStrip.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibilityObject | Obtiene AccessibleObject asignado al control. |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AllowDrop | Obtiene o establece un valor que indica si las operaciones de arrastrar y colocar y la reordenación de elementos se controlan mediante eventos que se implementan. |
| AllowItemReorder | Obtiene o establece un valor que indica si la clase ToolStrip controla la organización de elementos y las operaciones de arrastrar y colocar de forma privada |
| Anchor | Obtiene o establece los bordes del contenedor al que está enlazado un ToolStrip y determina cómo se cambia el tamaño de un ToolStrip con su elemento primario. |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView(Control). |
| AutoSize | Obtiene o establece un valor que indica si el control cambia automáticamente de tamaño para mostrar todo su contenido. |
| **Métodos** | |
| AccessibilityNotifyClients  (AccessibleEvents, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AccessibilityNotifyClients  (AccessibleEvents, Int32, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AdjustFormScrollbars(Boolean) | Ajusta las barras de desplazamiento del contenedor en función de las posiciones actuales de los controles y del control que está seleccionado. |
| BeginInvoke(Delegate) | Ejecuta el delegado especificado de forma asincrónica en el subproceso donde se creó el identificador subyacente del control. |
| BringToFront() | Coloca el control al principio del orden Z |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control. |
| CreateAccessibilityInstance() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles. |
| CreateControlsInstance() | Crea una nueva instancia de la colección de controles para el control. |
| CreateGraphics() | Crea el Graphics para el control. |
| CreateHandle() | Crea un identificador para el control. |
| **Eventos de Control** | |
| AutoSizeChanged | Se produce cuando ha cambiado la propiedad AutoSize. |
| BackColorChanged | Se produce cuando cambia el valor de la propiedad BackColor. |
| BackgroundImageChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BackgroundImageLayoutChanged | Se produce cuando cambia la propiedad BackgroundImageLayout. |
| BeginDrag | Se produce cuando el usuario empieza a arrastrar el control ToolStrip. |
| BindingContextChanged | Se produce cuando cambia el valor de la propiedad BindingContext. |
| CausesValidationChanged | Se produce cuando cambia la propiedad CausesValidation. |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado. |
| Click | Se produce cuando se hace clic en el control. |
| ClientSizeChanged | Se produce cuando cambia el valor de la propiedad ClientSize. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control ToolStripContainer.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AccessibilityObject | Obtiene AccessibleObject asignado al control. |
| AccessibleDefaultActionDescription | Obtiene o establece la descripción de la acción predeterminada del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleDescription | Obtiene o establece la descripción del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleName | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AccessibleRole | Obtiene o establece el nombre del control que las aplicaciones cliente de accesibilidad utilizan. |
| AllowDrop | Obtiene o establece un valor que indica si las operaciones de arrastrar y colocar y la reordenación de elementos se controlan mediante eventos que se implementan. |
| ActiveControl | Obtiene o establece el control activo en el control contenedor. |
| Anchor | Obtiene o establece los bordes del contenedor al que está enlazado un ToolStrip y determina cómo se cambia el tamaño de un ToolStrip con su elemento primario. |
| AutoScrollOffset | Obtiene o establece a dónde se desplaza este control en ScrollControlIntoView(Control). |
| AutoSize | Obtiene o establece un valor que indica si el control cambia automáticamente de tamaño para mostrar todo su contenido. |
| **Métodos** | |
| AccessibilityNotifyClients  (AccessibleEvents, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AccessibilityNotifyClients  (AccessibleEvents, Int32, Int32) | Notifica a las aplicaciones cliente de accesibilidad los objetos AccessibleEvents especificados del control secundario especificado. |
| AdjustFormScrollbars(Boolean) | Ajusta las barras de desplazamiento del contenedor en función de las posiciones actuales de los controles y del control que está seleccionado. |
| BeginInvoke(Delegate) | Ejecuta el delegado especificado de forma asincrónica en el subproceso donde se creó el identificador subyacente del control. |
| BringToFront() | Coloca el control al principio del orden Z |
| Contains(Control) | Obtiene un valor que indica si el control especificado es un control secundario del control. |
| CreateAccessibilityInstance() | Fuerza la creación del control visible, incluidos el identificador y los controles secundarios visibles. |
| CreateControlsInstance() | Crea una nueva instancia de la colección de controles para el control. |
| CreateGraphics() | Crea el Graphics para el control. |
| CreateHandle() | Crea un identificador para el control. |
| **Eventos de Control** | |
| AutoSizeChanged | Se produce cuando ha cambiado la propiedad AutoSize. |
| BackColorChanged | Se produce cuando cambia el valor de la propiedad BackColor. |
| BackgroundImageChanged | Se produce cuando cambia el valor de la propiedad BackgroundImage. |
| BackgroundImageLayoutChanged | Se produce cuando cambia la propiedad BackgroundImageLayout. |
| AutoValidateChanged | Se produce cuando cambia la propiedad AutoValidate. |
| BindingContextChanged | Se produce cuando cambia el valor de la propiedad BindingContext. |
| CausesValidationChanged | Se produce cuando cambia la propiedad CausesValidation. |
| ChangeUICues | Se produce cuando cambian las guías de la interfaz de usuario para el foco o el teclado. |
| Click | Se produce cuando se hace clic en el control. |
| ClientSizeChanged | Se produce cuando cambia el valor de la propiedad ClientSize. |

1. **Describa al menos 10 propiedades, 10 métodos y 10 eventos del control Timer.**

|  |  |
| --- | --- |
| **Propiedades** | |
| AutoReset | Obtiene o establece un valor booleano que indica si Timer debe generar el evento Elapsed solo una vez (false) o repetidamente (true). |
| CanRaiseEvents | Obtiene un valor que indica si el componente puede generar un evento. |
| Container | Obtiene IContainer que contiene Component. |
| DesignMode | Obtiene un valor que indica si Component está actualmente en modo de diseño. |
| Enabled | Obtiene o establece un valor que indica si Timer debería provocar el evento Elapsed. |
| Events | Obtiene la lista de controladores de eventos asociados a Component. |
| Interval | Obtiene o establece el intervalo, expresado en milisegundos, en el que se generará el evento Elapsed. |
| Site | Obtiene o establece el sitio que enlaza la clase Timer a su contenedor en modo de diseño. |
| SynchronizingObject | Obtiene o establece el objeto que se utiliza para calcular las referencias de las llamadas del controlador de eventos que se emiten al transcurrir un intervalo. |
|  |  |
| **Métodos** | |
| BeginInit() | Comienza la inicialización en tiempo de ejecución de un objeto Timer que se utiliza en un formulario o que lo utiliza otro componente. |
| Close() | Libera los recursos utilizados por Timer. |
| Dispose() | Libera todos los recursos que usa Component. |
| EndInit() | Finaliza la inicialización en tiempo de ejecución de un objeto Timer que se utiliza en un formulario o que utiliza otro componente. |
| GetType() | Obtiene el Type de la instancia actual. |
| Equals(Object) | Determina si el objeto especificado es igual al objeto actual. |
| MemberwiseClone() | Crea una copia superficial del Object actual. |
| Start() | Empieza a provocar el evento Elapsed estableciendo Enabled en true. |
| Stop() | Deja de desencadenar el evento Elapsed estableciendo el valor de Enabled en false. |
| ToString() | Devuelve una String que contiene el nombre del Component, si existe. Este método no se debe invalidar. |
| **Eventos de Control** | |
| Disposed | Se produce cuando el componente se elimina mediante una llamada al método Dispose(). |
| Elapsed | Se produce cuando transcurre el intervalo. |

1. **¿Para qué se usa el método Parse cuando se lo utiliza por ejemplo como Decimal.Parse?**

Se utiliza para convertir un número a su decimal equivalente

1. **¿Qué tipos de datos podemos manejar, qué rango de valores puede manejar cada uno y cuanto espacio ocupa cada uno?**
   1. Tipos simples
      1. Entero con signo:
         1. sbyte: 8 bits, de -128 a 127
         2. short: 16 bits, de -32,768 a 32,767
         3. int: 32 bits, de -2,147,483,648 a 2,147,483,647
         4. long: 64 bits, de -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807
      2. Entero sin signo:
         1. Byte: 8 bits, de 0 a 255
         2. Ushort: 16 bits, de 0 a 65,535
         3. Uint: 32 bits, de 0 a 4,294,967,295
         4. Ulong: 64 bits, de 0 a 18,446,744,073,709,551,615
      3. Caracteres Unicode: char:
      4. Punto flotante de IEEE:
         1. Float: 32 bits, de 1.5 × 10−45 a 3.4 × 1038, precisión de 7 dígitos
         2. Double: 64 bits, de 5.0 × 10−324 a 1.7 × 10308, precisión de 15 dígitos
      5. Decimal de alta precisión: decimal: 128 bits, al menos de –7.9 × 10−28 a 7.9 × 1028, con una precisión mínima de 28 dígitos
      6. Booleano: bool
   2. Tipos de enumeración
      1. Tipos definidos por el usuario con el formato enum E {...}
   3. Tipos de estructura
      1. Tipos definidos por el usuario con el formato struct S {...}
   4. Tipos de valor que aceptan valores NULL
      1. Extensiones de todos los demás tipos de valor con un valor null
2. **¿Cómo se declara un vector dinámico de tipo Integer?**

Se declara “int[] nombre vector”. Luego se debe inicializar el valor.

1. **¿Cómo se declara un vector de 10 posiciones de tipo Decimal?**

Se declara “deci[] vector= new vector[10]”

1. **¿Cómo se declara una matriz dinámica de tipo String?**

Se puede declarar “ string [ ][ ] matriz = new string [3][3];”

1. **¿Cómo se declara un array tridimensional dinámico de tipo Double?**

double[, ,] array3D = new double[1, 1, 1 ]

1. **¿Cuándo hablamos de boxing y unboxing a que nos referimos?**

Boxing es el proceso de convertir un valor al tipo object o a alguna interface implementada por este tipo de valor.

Unboxing es la conversión explícita del tipo object al tipo de valor o hacia un tipo de valor que implemente la interface

1. **¿Qué ámbitos existen para asignarle a las variables, los procedimientos y los métodos?**

Los ámbitos que existen para declarar variables, métodos y procedimientos son:

* Private
* Public

1. **¿Cómo se declara un procedimiento? Ejemplo.**

Se declara por la declaración private, public o internal.

Ejemplo:

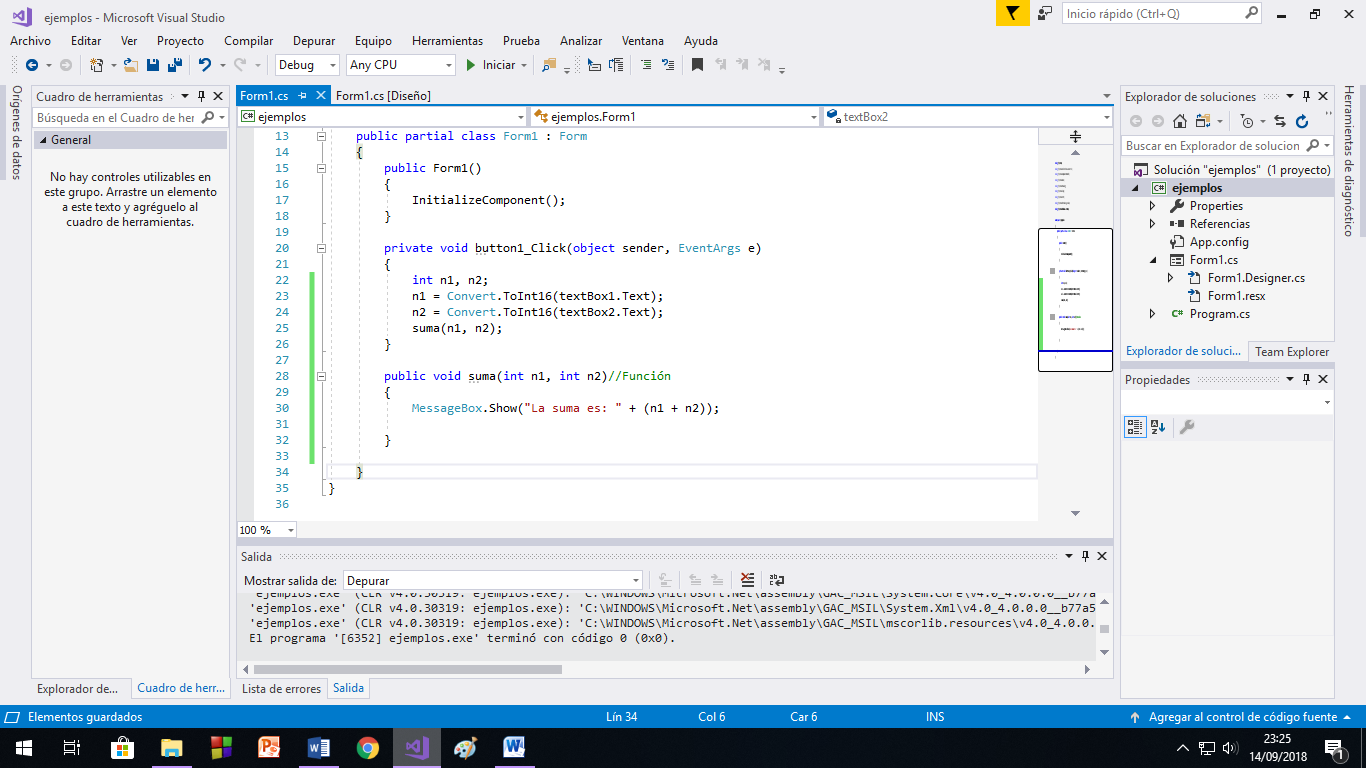
Private void HolaMundo()

{

MessageBox.Show(“Hola Mundo”);

}

1. **¿Cómo se declara una función? Ejemplo**



1. **¿Qué diferencia existe entre una función y un procedimiento?**

La principal diferencia entre ambas es que la función nos retorna un valor que resulta de ella y los procedimientos no

1. **¿Qué significa que el parámetro de un procedimiento se pasa por valor?**

Pasar un parámetro por valor con la intención de enviar la copia, pero no modificar el valor

1. **¿Qué significa que el parámetro de una función se pasa por referencia?**

El paso de parámetros por referencia permite a los miembros de funciones, métodos, propiedades, indexadores, operadores y constructores cambiar el valor de los parámetros y hacer que ese cambio persista en el entorno de la llamada

1. **Enumere las estructuras de decisión. Ejemplifique cada una de ellas y explique en qué se diferencian.**
   1. If
   2. If/Else
   3. If Anidados
   4. Switch
2. If: Si la condición es verdadera, entonces la instrucción se ejecuta; en este caso, condición debe ser una expresión que, una vez evaluada, debe devolver una booleana true o false. Con esta sintaxis, sólo la instrucción colocada después del if, se ejecutará si la condición es verdadera.

If( valor == 0)

{

MessageBox.Show(“El número ingresado fue 0”);

}

1. If / else: Al igual que el punto anterior, si la condición es verdadera, entonces la instrucción dentro de If se ejecuta; pero si la condición es falsa, se ejecutará la instrucción dentro del else.

If( valor == 0)

{

MessageBox.Show(“El número ingresado fue 0”);

}

Else

{

MessageBox.Show(“El número ingresado no fue 0”);

}

1. If Anidados: Si la condición es verdadera, se ejecutará la instrucción, solo que en este caso la instrucción posee otro if dentro, y dentro de ese nuevo if podría haber otros if o no, a esto se lo denomina if anidado.

If( valor < 0)

{

If(valor >2)

{

MessageBox.Show(“El número ingresado fue 1”):

}

If(valor<2)

{

MessageBox.Show(“El número ingresado no fue 1”);

}

}

1. Switch: El valor de la variable se evalúa al principio de la estructura. Luego el valor obtenido se compara con el valor especificado en el primer los casos que el usuario desea desarrollar, realizando una versión más prolija de un conjunto de If.

Switch(respuesta)

{

Case “Si”:

MessageBox.Show(“Respondió sí”);

Break;

Case “No”:

MessageBox.Show(“Respondió no”);

Break;

Default:

MessageBox.Show(“Usted no respondió correctamente”);

Break;

}

1. **Enumere las estructuras de repetición. Ejemplifique cada una de ellas y explique en qué se diferencian.**
   1. For
   2. While
   3. Do/While
   4. Foreach
2. For: Los bucles for van asignando valores a una variable desde un valor inicial hasta un valor final, y cuando la variable contiene un valor que está fuera del intervalo el bucle termina. Veamos la sintaxis para hacernos mejor a la idea:

for (i=0;i<10;i++)

{

Resultado += Vector[i];

}

1. While: se repetirá mientras una condición determinada se cumpla, o sea, devuelva true.

Int i = 0

while (i<20)

{

MessageBox.Show(“Ingrese un número”);

i++;

}

1. Do / while: Tiene mucho que ver con los bucles while. La diferencia es que se ejecuta siempre al menos una vez, mientras que los bucles while, pueden no ejecutarse ninguna vez.

Int i = 0;

do

{

MessageBox.Show(“El contador se incrementará”);

I++;

}

while (I != 3);

1. Foreach: Se utiliza para recorrer la colección en iteración y obtener la información deseada, pero no se puede utilizar para agregar o quitar elementos de la colección de origen, ya que se pueden producir efectos secundarios imprevisibles.

int[] diasMes = {31, 28, 21};

foreach(int dias in diasMes)

{

Console.WriteLine("Dias del mes: {0}", dias);

}

1. **¿Cómo se crea una estructura en el entorno visto en clase?**

Lo que conocíamos en c cómo estructuras, se conoce como clases en c#. Por ende, para crear una estructura precisamos crear una clase.

1. **¿Qué es y que puede contener una Solución en el entorno de trabajo visto en clase?**

La solución es un archivo de texto con su propio formato único; por lo general no está diseñada para editarse manualmente.

Una solución tiene un archivo \*.suo asociado que almacena la configuración, las preferencias y la información de configuración para cada usuario que haya trabajado en el proyecto.

1. **¿Qué es y que puede contener un Proyecto en el entorno de trabajo visto en clase?**

Un proyecto contiene todos los archivos de código fuente, iconos, imágenes, archivos de datos y cualquier otra cosa que se compilará en un programa ejecutable o sitio web, etc. necesarios para realizar la compilación. Un proyecto también contiene toda la configuración del compilador y otros archivos de configuración que podrían ser necesarios en diversos servicios o componentes con los que el programa se comunicará.

En un sentido literal, un proyecto es un archivo XML (\*.vbproj, \*.csproj, \*.vcxproj) que define una jerarquía de carpetas virtuales junto con las rutas de acceso a todos los elementos "contiene" y todas las configuraciones de compilación. En Visual Studio, se usa el archivo de proyecto mediante el Explorador de soluciones para mostrar el contenido del proyecto y la configuración.

1. **¿Qué es un formulario?**

Un formulario es una superficie visual en la que se muestra información al usuario.

1. **¿Qué extensión posee el archivo que contiene información sobre la Solución?**

Posee una extensión .sln o .suo .

1. **¿Qué extensión posee el archivo que contiene información sobre el proyecto?**

Posee una extensión .csproj , vbproj o vcxproj .

1. **¿Qué extensión posee el archivo que contiene el código que Ud. Programó en VB.NET?**

Posee una extensión .resx .

1. **¿Qué carpeta contiene el archivo de la Solución?**

La carpeta .vs .

1. **¿Qué carpeta contiene el archivo del Proyecto?**

La carpeta con el nombre que el usuario eligió al momento de nombrar al proyecto.

1. **¿Qué carpeta contiene el código que Ud/ programó en el formulario?**

La carpeta con el nombre que el usuario eligió al momento de nombrar al proyecto.

1. **Enumere, explique para sirve y de un ejemplo de 10 funciones que permitan manipular datos del tipo String.**
   1. Clone(): Devuelve una referencia a la instancia de String.
   2. Compare(String, Int32, String, Int32, Int32): Compara subcadenas de dos objetos String especificados y devuelve un entero que indica su posición relativa en el criterio de ordenación.
   3. Concat(IEnumerable<String>): Concatena los miembros de una colección IEnumerable<T> construida de tipo String.
   4. Copy(String): Crea una nueva instancia de String con el mismo valor que una String especificada.
   5. EndsWith(String): Determina si el final de esta instancia de cadena coincide con la cadena especificada.
   6. Equals(Object): Determina si esta instancia y un objeto especificado, que también debe ser un objeto String, tienen el mismo valor.
   7. Format(String, Object): Reemplaza uno o más elementos de formato de una cadena por la representación de cadena de un objeto especificado.
   8. GetType(): Obtiene el Type de la instancia actual.
   9. IndexOf(Char) : Devuelve el índice de base cero de la primera aparición del carácter Unicode especificado en esta cadena.
   10. Insert(Int32, String): Devuelve una nueva cadena en la que se inserta una cadena especificada en una posición de índice concreta en esta instancia.
2. **Enumere, explique para sirve y de un ejemplo de 10 funciones que permitan manipular datos del tipo date.**
   1. AddMonths(Int32): Añade un número específico de meses.
   2. Compare(DateTime, DateTime): Compara ambos DateTime y devuelve la diferencia como entero.
   3. DaysInMonths(Int32, Int32): Devuelve la cantidad de días en meses y año.
   4. ToLocalTime(): Convierte el valor de el objeto DateTime actual al valor local.
   5. ToLongDateString(): Convierte el valor de el objeto DateTime a un valor equivalente y largo en string.
   6. ToShortDateString(): Convierte el valor de el objeto DateTime a un valor equivalente y corto en string
   7. ToUniversalTime(): Convierte el valor de DateTime coordinado con el tiempo Universa (UTC).
   8. AddYears(Int32): Añade un número específico de años.
   9. AddMilliseconds(Int32): Añade un número específico de milisegundos.
   10. AddDays(Int32): Añade un número específico de días.
3. **Enumere, explique para sirve y de un ejemplo de 10 funciones matemáticas**
   1. Abs(Decimal): Devuelve el valor absolute de un número decimal.
   2. Ceiling(Decimal): Devuelve el valor entero más chico que sea mayor o igual al número decimal nombrado.
   3. Exp(Double): Devuelve e elevado a la potencia nombrada.
   4. Floor(Decimal): Devuelve el valor entero más grande que sea menor o igual al decimal nombrado
   5. Log10(Double): Devuelve el logaritmo con base 10 del número nombrado.
   6. Pow(Double, Double) : Devuelve un número elevado a cierta potencia.
   7. Round(Decimal, Int32, MidpointRounding): Redondea el valor de un decimal a un número de dígitos fraccionarios. Un parámetro especifica cómo redondear el valor si está en medio de dos números.
   8. Sqrt(Double): Devuelve un número con cierta raiz.
   9. Tan(Double): Devuelve la tangente de un ángulo.
   10. Truncate(Decimal): Calcula la parte entera de un número decimal.
4. **¿Qué es una propiedad?**

Son los aspectos de los campos y los métodos.

1. **¿Qué es un método?**

Son las funciones que representan las acciones que van a hacer los elementos

1. **¿Qué es un evento?**

Son las reacciones que los elementos pueden tener

1. **Dado un procedimiento cómo hace para asociarlo a que se ejecute cuando se da un evento**

Para asociar al procedimiento con el evento, solo hay que ir al código del evento e incluirle el procedimiento.

1. **¿Un procedimiento puede estar asociado a más de un evento?**

Si. Un procedimiento puede ser utilizado en múltiples eventos y funciones.

1. **Dado un procedimiento asociado a un evento que representa el parámetro sender.**

Sender representa una referencia al objeto que lanza ese evento

1. **¿Por qué el parámetro sender es del tipo Object?**

Porque al ser del tipo Object porque hace referencia a un objeto que desencadena un evento.

1. **Dado un procedimiento asociado a un evento que representa el parámetro e.**

Sender representa una referencia al objeto que lanza ese evento

1. **¿Qué tipo posee el parámetro e de un procedimiento asociado a un evento?**

Su tipo es Object.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede abrir una solución nueva o una existente.**

Menú Archivo>Abrir>Proyecto o solución>Seleccionamos el archivo y presionamos click en abrir.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede obtener la ventana de Error List.**

Menú Ver>Error List.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede cambiar las propiedades del proyecto.**

Desde el menú del Explorador de soluciones.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede obtener la ventana Toolbox.**

Menú Ver > Toolbox.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede colocar puntos de interrupción.**

Desde el menú Depura > Nuevo punto de interrupción.

1. **Tomando como referencia el entorno de programación utilizado en clase, desde que menú puede obtener la ventana Solution Explorer.**

Menú Ver > Solution Explorer.

1. **¿Para qué se utiliza null?**

Se utiliza para establecer que un dato está vacío.

1. **¿Si me encuentro en el Formulario llamado Form1, a quién hace referencia this?**

“This” hace referencia al formulario en el que me encuentro, es decir, Form1

1. **Enumere y explique los operadores aritméticos.**
   1. +: representa la suma de dos o más valores o variables
   2. –: representa la resta de dos o más valores o variables
   3. \*: representa la multiplicación de dos o más valores o variables
   4. /: representa la división de dos o más valores o variables
   5. %: representa el módulo de dos o más valores o variables
2. **Enumere y explique los operadores de asignación.**
   1. +=: asignación y suma
   2. -=: asignación y resta
   3. \*=: asignación y multiplicación
   4. /=: asignación y división
   5. %=: asignación y módulo
3. **Enumere y explique los operadores de comparación.**
   1. ==: igual que
   2. !=: diferente que
   3. >: mayor que
   4. <: menor que
   5. >=: mayor igual que
   6. <=: menor igual que
4. **Enumere y explique los operadores de concatenación**
   1. +: agrega detrás del “+” lo que le sigue
   2. &: se define sólo para string, y es recomendable su uso para concatenar cadenas
5. **Además de los operadores explicados previamente, existen más operadores, en caso afirmativo enumérelos y explique para que sirven.**

Sí, existen los operadores lógicos. Son aquellos que permite obtener únicamente dos respuestas, sí o no.

1. &&: AND
2. ||: OR
3. !: NOT (negación)
4. **¿Qué es y para qué se utilizan las enumeraciones?**

Son un tipo especial de variables que tienen la propiedad de que su rango de valores es un conjunto de constantes entera denominadas constantes de enumeración, a cada una de las cuales se le asigna un valor.

GUÍA DE TRABAJOS PRÁCTICOS

UNIDAD I

Introducción a la programación visual I

1. Desarrollar un programa que permita ingresar dos números por medio de inputbox

y retorne la suma de los mismos en un messagebox.

2. Desarrollar un programa que permita ingresar dos números por medio de cajas de

texto y retorne la suma de los mismos en un label.

3. Desarrollar un programa que permita cargar una cantidad n de números ingresados

por el usuario y mostrar los números ingresados en un textbox y el resultado en un

label.

4. Desarrollar un programa que permita cargar números hasta que el usuario indique

lo contrario. Mostrar los números ingresados en un textbox y el resultado en un

label.

5. Desarrollar un programa que permita calcular la suma, resta, multiplicación,

división, potencia y raíz cuadrada de dos números. Seleccione los controles que

utilizará para ingresar los números y visualizar el resultado. Las operaciones se

ejecutarán con botones de comando.

6. Desarrollar un programa que permita calcular la suma, resta, multiplicación,

división, potencia y raíz cuadrada de los números ingresados por el usuario hasta

que este decida no ingresar más. Seleccione los controles que utilizará para ingresar

los números y visualizar el resultado. Las operaciones se ingresan secuencialmente

comenzando con un número, una operación, un número etc.

7. Desarrollar un programa que le permita al usuario indicar cuantos números desea

ingresar. Luego mostrar los pares en una caja de texto y los impares en otra.

8. Desarrollar un programa que le permita al usuario ingresar números hasta que

indique lo contrario. Luego mostrar los pares en una caja de texto y los impares en

otra.

9. Desarrollar un programa que le permita al usuario indicar un número mínimo y un

número máximo. Mostrar en un Listbox todos aquellos que son primos.

10. Desarrollar un programa que permita calcular la serie de fibonacci hasta un número indicado por el usuario.

11. Desarrollar un programa que permita calcular los primeros n números de la serie de Fibonacci. El número n es ingresado por el usuario.

12. Desarrollar un programa que permita calcular los intereses de un plazo fijo. Para ello nos informan que la manera de obtenerlo es haciendo M \* T \* D / 36500.

Donde: M=Monto, T=Tasa Nominal Anual, D=Días que dura la imposición. Mostrar

los valores ingresados y los resultados en una Grilla del tipo DataGridView.

13. Desarrollar un programa que simule la tirada de un dado.

14. Desarrollar un programa que simule la tirada de dos dados y mostrar en una grilla

la cantidad de veces que salió cada número y el porcentaje que representa esa

cantidad en el total de tiradas. Las tiradas se deben realizar al oprimir un botón.

15. Tomar como base el ejercicio del punto 14 y automatizar la tirada de dados con un

timer. Que al timer se le pueda configurar desde la interfaz la velocidad con que se

tiran los dados.

16. Desarrollar un programa que permita transformar temperaturas entre las

siguientes escalas: Celsius, Fahrenheit, Kelvin, Rankine. el programa deberá

permitir indicar: la escala inicial, la escala final y la cantidad a transformar. Se debe

obtener la cantidad en la escala final. Los valores ingresados y calculados deberán

permaneces visibles en una grilla.

17. Desarrollar un programa que permita transformar unidades de tiempo. La unidad

de tiempo origen se podrá seleccionar mediante controles RadioButtton y las

unidades de salida por medio de CheckBox. Decida Ud. la Interfaz más adecuada

para visualizar los resultados. Las unidades a considerar son:

Horas 1 x 3600 s

Minuto 1 x x60 s

Segundo 1 s

Decisegundo 1 × 10-1 s

Centisegundo 1 × 10-2 s

Milisegundo 1 × 10-3 s

Microsegundo 1 x 10-6 s

Nanosegundo 1 x 10-9 s

Picosegundo 1 × 10–12 s

Femtosegundo 1 × 10–15 s

Attosegundo 1 × 10–18 s

Zeptosegundo 1 × 10–21 s

Voctosegundo 1 × 10–24 s

18. Desarrollar un programa que permita transformar unidades de medida de la

información. La unidad de medida de la información origen se podrá seleccionar

mediante controles RadioButtton y las unidades de salida por medio de CheckBox.

Decida Ud. la Interfaz más adecuada para visualizar los resultados. Las unidades a

considerar son:

bit

Byte

KiloByte

MegaByte

GigaByte

TeraByte

PetaByte

ExaByte

ZetaByte

YottaByte

BrontoByte

GeopByte

19. Desarrolle un programa que simule el juego del TA-TE-TI

20. Desarrolle un programa utilizando el control ColorDialog y que el color seleccionado se le asigne al color de fondo del formulario.

21. Desarrolle un programa utilizando el control FileDialog y seleccionar una imagen

que se vea en un control PictureBox.