

Programación Visual Basic .NET 2010

Índice

Tema 3: Aplicaciones Windows en Visual Basic .NET 2010

- 3.1 Creando una Aplicación a Nivel de Windows
 - 3.2 Componentes más Comunes
 - 3.3 Metodología de Solución
-

Objetivos:

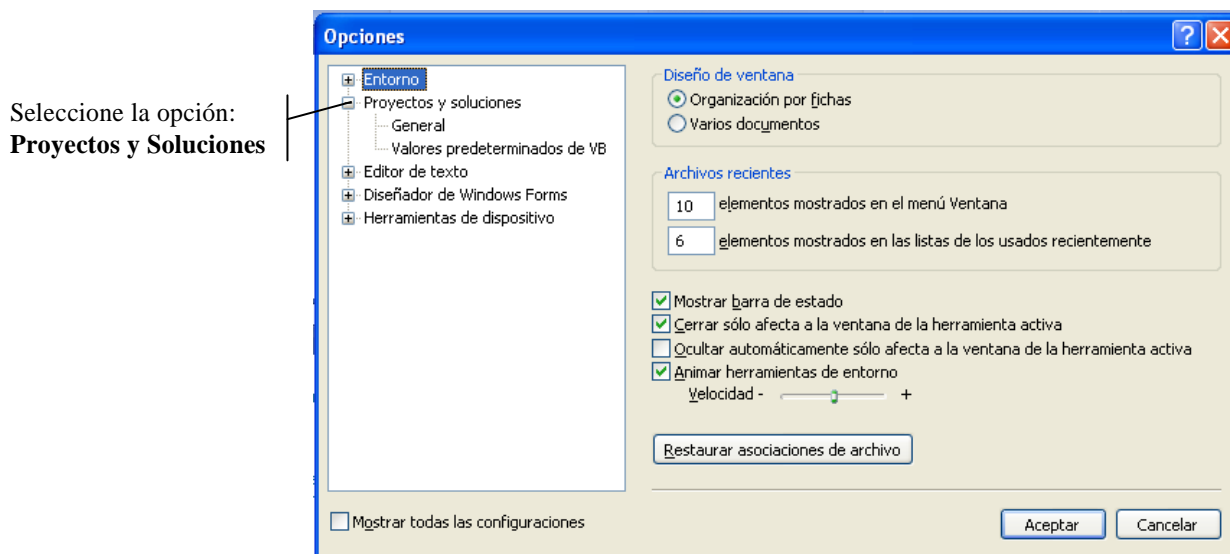
- ❖ Conocer los componentes más comunes y usados en una Aplicación a nivel de Windows.
- ❖ Conocer los Procedimientos Básicos que se utilizan para la Construcción de Aplicaciones a nivel Windows.

Temática:

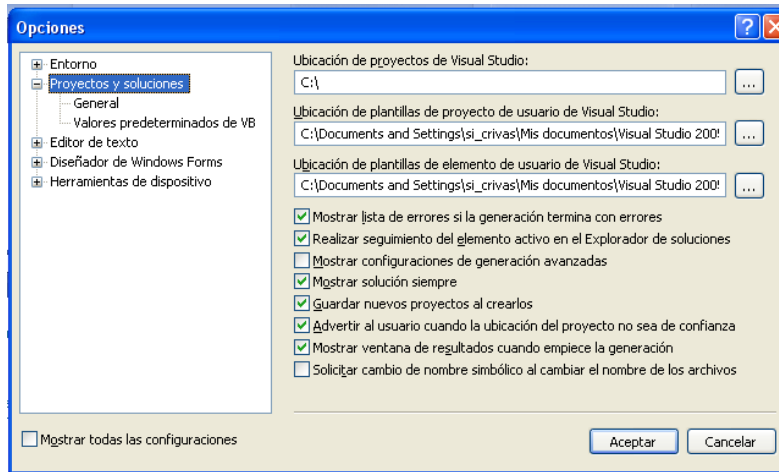
3.1- Creando una Aplicación a Nivel de Windows

Toda aplicación a nivel de Windows en Visual Basic .NET 2010 se seguirá los pasos siguientes:

- ❖ **Configuración Inicial para toda Aplicación en Visual Basic .NET 2010**
 - Cargue en la memoria el programa Visual Basic .NET 2010. Para ello, de clic en el botón “Inicio”, seleccione la opción “Programas”, seleccione la opción “Microsoft Visual Studio 2010” y finalmente seleccione la opción “Microsoft Visual Studio 2010”.
 - Seleccione la opción **Herramientas** del menú principal del programa.
 - Seleccione la opción **Opciones**.
 - Aparecerá la ventana siguiente:



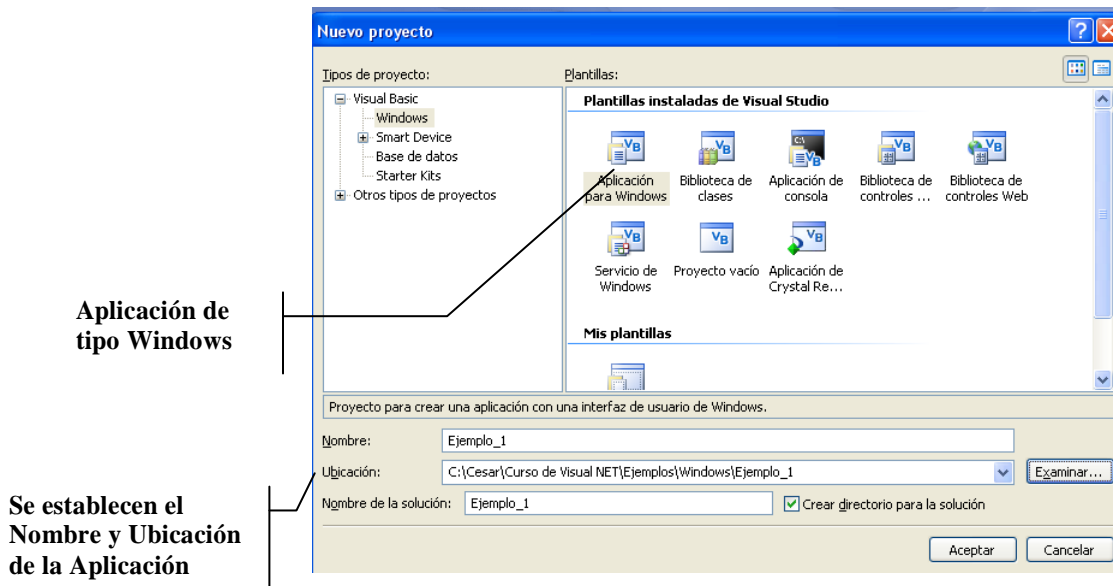
- Del listado de opciones presentadas en la **Vista Jerárquica** (ubicada en la parte izquierda), ubíquese en la opción **Proyectos y Soluciones** dando clic sobre él. Configurarlos de tal forma que quede así:



- De clic en el botón **Aceptar** para guardar los cambios. Con esto se deja definido al menos, que cuando usted desarrolle una aplicación le pida la ubicación para guardarlo.

❖ **Aplicación de Tipo Windows**

- Crear una **Aplicación de tipo Windows**. Para ello, Seleccione la opción “**Archivo**” del Menú Principal de **Visual Basic .NET 2010** y seleccione la opción “**Nuevo Proyecto**”. Le abrirá la ventana de **Nuevo Proyecto** indicando qué tipo de proyecto va a realizar. De acuerdo a la ventana, seleccione el tipo de proyecto “**Aplicación para Windows**”. Además, guardar la aplicación con el nombre de **Ejemplo_1** ubicado en la **unidad C** de su computadora.



- Finalmente, se da clic en el botón **Aceptar**.

- Aparecerá el siguiente Entorno de Trabajo, similar al de **Visual Basic 6.0**:

Icono que permite visualizar la **Ventana de Explorador de Soluciones**.

Icono que permite visualizar la **Ventana de Propiedades**.

Ventana de Explorador de Soluciones mostrando los formularios, módulos y otros archivos del proyecto.

Ventana de Propiedades mostrando las propiedades de cada objeto u componente de la aplicación.

Es la **Barra de Herramientas** o de **Componentes** equivalente a la de Visual Basic 6.0

Ventana de Resultados cuando es compilada la aplicación para detectar errores.

Formulario por Omisión objeto principal donde se ubicarán todos los objetos visuales necesarios para la aplicación.

3.2- Componentes más Comunes

Objeto: Formulario

Definición

En **Visual Basic .NET 2010** para una aplicación de tipo Windows comúnmente trabaja con **Formularios**. Un **Formulario** es una ventana muy importante ,porque aquí se dibujan o pegan todos los controles que necesitamos utilizar en nuestro programa o aplicación. Se pueden utilizar tantos formularios como se necesiten, y dependiendo de la utilidad que se les dé, estos serán de diferentes tipos. Así, se puede crear un formulario que contenga un gráfico, para visualizar información o para ingresar datos.

Tipos:

Hay cuatro tipos de formularios, que son los más usados, como son:

- ❖ **El formulario MDI**, que es un contenedor de otros formularios, por ejemplo tenemos las ventanas de Word que es un formulario MDI, ya que contiene o puede contener diversas ventanas de documentos abiertos.

- ❖ **Los formularios Child**, que son aquellos formularios hijos, en otras palabras son todos aquellos formularios que van a ser contenidos en el **formulario MDI**.
- ❖ **El formulario Normal o el de Omisión**, que es el que más vamos a usar, y el más común.
- ❖ **El formulario Parents**, que es aquel formulario que va a contener todos los controles de otro formulario, por ejemplo, si tenemos un formulario llamado frmPadre, que contiene un menú y un botón grabar y creamos un nuevo formulario de tipo parents llamado frmHijo y le especificamos en su propiedad de parents el formulario frmPadre, automáticamente el formulario frmHijo aparecerá con el menú y el botón grabar.

Propiedades más usados en los Formularios:

| Propiedad | Descripción |
|-----------------------------|---|
| (Name) | Permite identificar el formulario, su prefijo de cualquier formulario es frm |
| AccepButton | Si esta establecido, el botón se activa cuando el usuario presiona la tecla [Enter] ([INTRO]) |
| AccesibleDescription | Descripción que se proporciona a clientes con accesibilidad. |
| AccesibleName | Nombre que se proporciona a los clientes con accesibilidad. |
| AccesibleRole | Función que se proporciona a clientes con accesibilidad. |
| AllowDrop | Determina si el control recibe notificaciones de arrastrar y colocar. |
| AutoScaleMode | Determina cuando el formulario o el control, cuando la escala de la resolución de la pantalla o de la fuente de letra a cambiado. |
| AutoScroll | Determina si aparecen automáticamente las barras de desplazamiento cuando los controles están situados fuera del área de clientes del formulario. |
| AutoScrollMargin | Establece el margen alrededor del control durante el AutoScroll. |
| AutoScrollMinSize | Tamaño lógico mínimo para la zona de desplazamiento automático. |
| AutoSize | Establece si el formulario se adecua al tamaño especificado de los controles. |

Propiedades más usados en los Formularios:

| | |
|------------------------------|---|
| AutoSizeMode | Especifica el modo del tamaño del formulario |
| BackColor | Color de fondo a utilizar para mostrar texto o gráficos en el control. |
| BackgroundImage | Imagen de fondo utilizada para el formulario. |
| BackgroundImageLayout | Especifica como se presentara la imagen de fondo del formulario, puede ser en mosaico, estirado, centrada. |
| CancelButton | Si esta establecido, el botón se activa cuando el usuario presiona la tecla ESC |
| CausesValidation | Indica si este control causa y genera eventos de validación. |
| ContextMenuStrip | Indica el menú contextual que se muestra cuando el usuario hace clic derecho. |
| ControlBox | Determina si el formulario tiene un cuadro de menú control el que se encuentra en la esquina superior derecha de la ventana. |
| Cursor | Indica el tipo de cursor que aparece al pasar el Mouse por el control |
| Enabled | Indica si el control esta habilitado |
| Font | Indica la fuente utilizada para mostrar texto en el control. |
| ForeColor | Indica el color de primer plano utilizado para mostrar texto o gráficos en el control. |
| FormBorderStyle | Controla la apariencia del borde del formulario. Esto también influye en como se muestra la barra de titulo y que botones pueden aparecer en ella. |
| HelpButton | Determina si un formulario tiene botón de ayuda en la barra de titulo. |
| Icon | Indica el icono para un formulario. Esto se muestra en el cuadro de menú del formulario del sistema y cuando el formulario se minimiza, Width: Ancho, Height: Altura. |
| ImeMode | Determina el estado IME (Editor de Métodos de Entrada) del objeto cuando se selecciona. |
| IsMDIContainer | Determina si el formulario es un contenedor MDI. |
| KeyPreview | Determina si los eventos de teclado para los controles del formulario se registran junto con el formulario. |
| Language | Indica el idioma traducido actualmente. |
| Localizable | Determina si se va a generar el código localizable para este objeto. |
| Location | Posición de la esquina superior izquierda del control, respecto al contenedor. |
| Locked | Determina si se puede mover o cambiar de tamaño el control. |
| MainMenuStrip | Indica el menú principal del formulario. Debe establecerse en un componente de tipo MainMenuStrip. |

Propiedades más usados en los Formularios:

| | |
|------------------------|--|
| MaximizeBox | Determina si el formulario contendrá el botón de maximizar. |
| MaximunSize | Indica el tamaño máximo de la ventana al maximizar. |
| MinimizeBox | Determina si el formulario contendrá el botón de minimizar. |
| MinimunSiza | Indica el tamaño mínimo de la ventana al minimizar. |
| Opacity | Determina la opacidad o transparencia del formulario; 0% es transparente, 100% es opaco (visible) |
| Padding | Indica el espacio entre los controles que va a contener el formulario. |
| RightToLeft | Indica si el control debe dibujarse de derecha a izquierda para idiomas que se escriben en esa dirección. |
| ShowIcon | Determina si se muestra el icono del formulario en la barra de título. |
| ShowInTaskBar | Determina si el formulario aparece en la barra de tareas de Windows. |
| Size | Determina el tamaño del formulario o control en píxeles. |
| SizeGripStyle | Determina cuando se muestra SizeGrip en un formulario. |
| StarPosition | Determina la posición del formulario cuando aparece por primera. |
| Tag | Datos definidos por el usuario asociado con el control. |
| Text | Muestra el texto contenido en el formulario en la barra de título. |
| TopMost | Determina si el formulario está sobre todos los demás formularios que no son de nivel superior, incluso cuando esta desactivado. |
| TransparencyKey | Indica el color que parece transparente cuando se aplica al formulario. |
| UseWaitCursor | Indica si se muestra un cursor de espera. |
| WindowsState | Determina el estado visual inicial del formulario, puede ser minimizado, maximizado o normal. |

Métodos más usados en los Formularios:

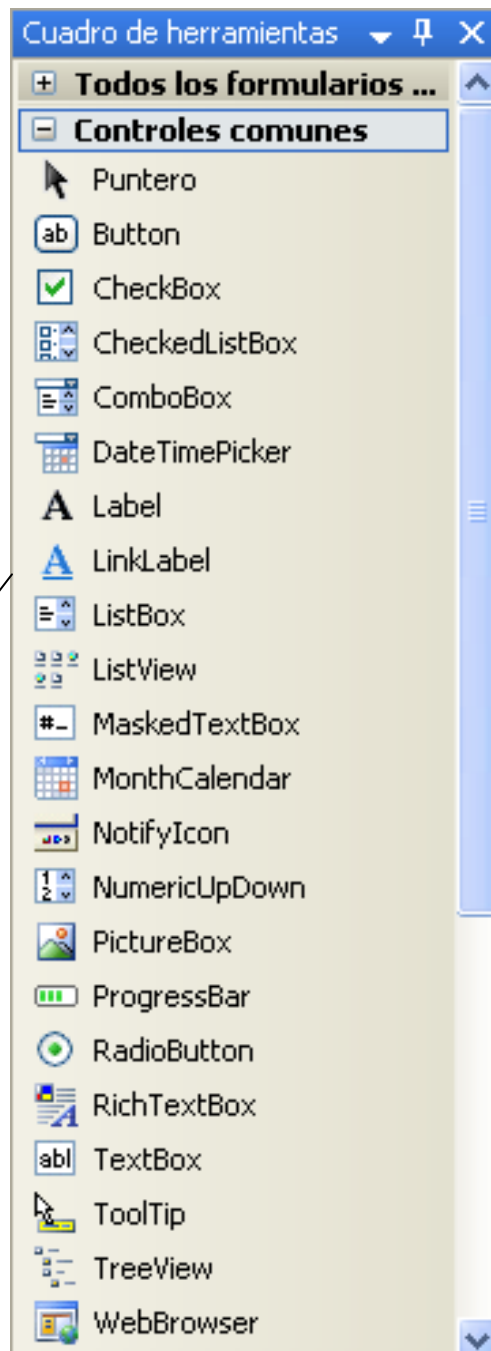
| Método | Descripción |
|------------------------|--|
| Activate | Activa el formulario y le otorga el foco |
| Close | Cierra el formulario |
| CreateControl | Obliga a que se cree el control, incluidos el identificador y los controles secundarios. |
| CreateGraphics | Crea el objeto Graphics para el control |
| Dispose | Sobrecargado. Libera los recursos utilizados por componentes. |
| FindForm | Recupera el formulario en el que se encuentra el control. |
| Finalize | Libera todos los recursos del formulario y del sistema. |
| Focus | Establece el foco de entrada al control. |
| GetType | Obtiene el objeto Type de la instancia actual. |
| Hide | Oculto el formulario o control al usuario. |
| Invalidate | Sobrecargado, Invalida una región específica del control y hace que se envíe un mensaje de dibujo al control. |
| LayoutMdi | Organiza los formularios MDI secundarios del formulario MDI principal. |
| Refresh | Obliga al control invalidar su área cliente y acto seguido obliga a que vuelva a dibujarse el control y sus controles secundarios. |
| RemoveOwnedForm | Quita un formulario con propietario de este formulario. |
| ResetText | Restablece el valor predeterminado de la propiedad Text. |
| Select | Sobrecargado, activa un control. |
| Show | Muestra el control o formulario al usuario. |
| ShowDialog | Sobrecargado. Muestra el formulario como un cuadro de diálogo modal. |
| Update | Hace que el control vuelva a dibujar las regiones no válidas en su área de cliente. |
| Validate | Valida el último control no validado y sus predecesores, pero sin incluir el control actual. |

Eventos más usados en los Formularios:

| Evento | Descripción |
|-------------------------------|--|
| Activated | Se produce cuando el formulario se activa por código de programa o por parte del usuario. |
| BackGroundImageChanged | Se produce cuando el valor de la propiedad BackGroundImage cambia. |
| Click | Se produce cuando el usuario hace clic en el control. |
| Disposed | Agrega un controlador de eventos para escuchar al evento Disposed en el componente. |
| DoubleClick | Se produce cuando se hace doble clic en el control. |
| EnabledChanged | Se produce cuando cambia el valor de la propiedad enabled. |
| Enter | Se produce cuando se entra en el control. |
| FormClosed | Tiene lugar cuando el formulario esta cerrado. |
| FormClosing | Tiene lugar cuando se cierra el formulario. |
| GotFocus | Se produce cuando el control recibe el foco. |
| Invalidated | Se produce cuando es necesario volver a dibujar un control. |
| KeyDown | Se produce cuando se presiona una tecla mientras el control tiene el foco. |
| KeyPress | Se produce cuando se presiona una tecla mientras el control tiene el foco. |
| KeyUp | Se produce cuando se suelta una tecla mientras el control tiene el foco. |
| Layout | Se produce cuando el foco de entrada deja el control |
| Leave | Se produce cuando el foco de entrada deja el control. |
| Load | Se produce antes de que se muestre un formulario por primera vez. |
| LostFocus | Se produce cuando el control pierde el foco. |
| MenuComplete | Se produce cuando el menú de un formulario pierde el foco. |
| MenuStart | Se produce cuando el menú de un formulario recibe el foco. |
| MouseDown | Se produce cuando el puntero del Mouse esta sobre el control y se presiona un botón del Mouse. |
| MouseEnter | Se produce cuando el puntero del Mouse entra en el control. |
| MouseHover | Se produce cuando el puntero del Mouse se sitúa encima del control. |
| MouseLeave | Se produce cuando el puntero del Mouse deja el control. |
| MouseMove | Se produce cuando el puntero del Mouse se mueve sobre el control. |
| MouseUp | Se produce cuando el puntero del Mouse esta encima del control y se suelta el botón del Mouse. |
| MouseWheel | Se produce cuando la rueda del Mouse se mueve mientras el control tiene el foco. |
| Move | Se produce cuando se mueve el control. |
| VisibleChanged | Se produce cuando cambia el valor de la propiedad visible. |


















Con la introducción del **.NET Framework 2.0** se cuenta con diversidades de controles y objetos visuales que podemos utilizar en nuestras aplicaciones de tipo Windows. Todos estos están ubicados en la **Barra de Herramientas o Componentes**:

Componente Comunes para toda Aplicación de Tipo Windows ubicado en Barra de Herramientas del Visual Basic .NET 2010





Ahora, en la siguiente tabla se definen los componentes que integran la **Barra de Herramientas o de Componentes**, definidos por **Categorías**:







Categoría No. 1: Componentes Comunes

| Icono | Nombre | Prefijo | Descripción |
|---|-----------------------|---------|---|
|  | Button | btn | Se utiliza para iniciar, detener o interrumpir un proceso. |
|  | CheckBox | chk | Muestra una casilla de verificación y una etiqueta para texto. Se utiliza n general para establecer opciones. |
|  | CheckedListBox | clb | Muestra una lista desplazable de elementos, cada uno acompañado por una casilla de verificación. |
|  | ComboBox | cbo | Muestra una lista desplegable de elementos. |
|  | DateTimePicker | dtp | Muestra un calendario gráfico desplegable, que permite que los usuarios seleccionen una fecha o una hora. |
| A | Label | lbl | Muestra texto que los usuarios no pueden modificar directamente. |
|  | LinkLabel | llbl | Muestra texto en forma de vínculo de estilo Web y desencadena un evento cuando el usuario hace clic en el texto. |
|  | ListBox | lst | Muestra una lista de textos, también llamados elementos. |
|  | ListView | lstv | Muestra los elementos ya sean texto solo o con gráficos (iconos). |
|  | MaskedTextBox | msk | Captura texto escrito por el usuario, ya sea teniendo un formato de entrada. |
|  | MonthCalendar | mnc | Muestra un calendario gráfico que permite que los usuarios seleccionen un intervalo de fechas. |
|  | NotifyIcon | nti | Muestra un icono, en el área de notificación de estado de la barra de tareas, que representa una aplicación que se ejecuta en segundo plano. |
|  | NumericUpDown | nud | Muestra una lista de número a través de la cual los usuarios se pueden desplazar mediante botones hacia arriba y hacia abajo. |
|  | PictureBox | pic | Muestra archivos de imágenes, tales como mapas bits e iconos, gifs, etc. |
|  | ProgressBar | pgb | Muestra una barra de progreso, mientras realiza una instrucción. |
|  | RadioButton | rdb | Muestra un botón que puede activarse o desactivarse. |
|  | RichTextBox | rtb | Habilita la presentación del texto con formato de texto sencillo o de texto enriquecido (RTF). |
|  | TextBox | txt | Muestra texto escrito en tiempo de diseño que puede ser editado por los usuarios en tiempo de ejecución. |
|  | ToolTip | ttp | Este control activa la propiedad ToolTip en los demás controles, su función es brindar una pequeña información al pasar el Mouse por el control que queramos. |
|  | TreeView | trv | Muestra una colección jerárquica de objetos de nodo que puede constar de texto con casilla de verificación o iconos opcionales. |
|  | WebBrowser | wbr | Permite crear un explorador de Internet con fácil navegación, también se puede abrir múltiples archivos. |





Categoría No. 2: Componentes

| Icono | Nombre | Prefijo | Descripción |
|---|------------------|---------|--|
|  | ImageList | img | Es un contenedor de imágenes, para que después se use con otros controles. |
|  | Timer | tmr | Sirve para realizar un conteo de tiempo, en tiempo de ejecución. |






Categoría No. 3: Componentes de tipo Contenedor

| Icono | Nombre | Prefijo | Descripción |
|---|-------------------------|---------|---|
|  | FlowLayoutPanel | flp | Es un contenedor de controles que no permite la modificación de la posición del control. |
|  | GroupBox | gpb | Agrupar un conjunto de controles (tales como botones de opción) en un marco con etiqueta. |
|  | Panel | pnl | Agrupar un conjunto de controles en un marco sin etiqueta que permite el desplazamiento. |
|  | SplitContainer | spc | Es un contenedor de controles con dos paneles, en donde se puede insertar cualquier tipo de control. |
|  | TabControl | tbc | Proporciona una página con fichas para organizar y tener acceso a controles agrupados de forma eficiente. |
|  | TableLayoutPanel | tlp | Es un organizador de controles, ya que podemos añadir filas o columnas para control que queramos. |






Categoría No. 4: Componentes para Acceso a Base de Datos

| Icono | Nombre | Prefijo | Descripción |
|---|-------------------------|---------|--|
|  | BindingNavigator | bgn | Este control nos permite la navegación de datos, conectada a una base de datos. |
|  | BindingSource | bgs | Este control nos permite la conexión y la navegación de datos, de una base de datos. |
|  | DataGripView | dgv | Nos permite visualizar los datos de una tabla o de una base de datos. |
|  | DataSet | dts | Nos permite la creación y la conexión de una base de datos. |

Categoría No. 5: Componentes de tipo Dialogo

| Icono | Nombre | Prefijo | Descripción |
|---|----------------------------|---------|---|
|  | ColorDialog | cdg | Muestra el cuadro de diálogo de selección de colore, que permite que los usuarios seleccionen el color de un elemento de la interfaz. |
|  | FolderBrowserDialog | fbd | Muestra un cuadro de diálogo que permite buscar o una carpeta de destino. |
|  | FontDialog | fdg | Muestra un cuadro de diálogo que permite que los usuarios establezcan una fuente y sus atributos. |
|  | OpenFileDialog | ofd | Muestra un cuadro de dialogo que permite que los usuarios se desplacen hasta un archivo y lo seleccionen. |
|  | SaveFileDialog | sfd | Muestra un cuadro de dialogo que permite que los usuarios guarden un archivo. |

Categoría No. 6: Componentes de tipo Menú

| Icono | Nombre | Prefijo | Descripción |
|---|---------------------------|---------|---|
|  | ContextMenuStrip | cms | Implementa un menú que aparece cuando el usuario hace clic en un objeto con el botón secundario del Mouse. |
|  | MenuStrip | mnu | Proporciona una interfaz en tiempo de diseño para la creación de menús. |
|  | StatusStrip | sts | Muestra información acerca del estado actual de la aplicación mediante una ventana con marco, habitualmente en parte inferior de un formulario. |
|  | ToolStrip | tls | Contiene una colección de botones, combos, etiquetas, etc. |
|  | ToolStripContainer | tsc | Muestra una interfaz de botones con menú. |

Las propiedades para cada uno de estos componentes lo aprenderemos a medida que las utilizaremos en los ejemplos correspondientes. El **Editor de Código** se invoca de la misma forma que se hacía en **Visual Basic 6.0** y tiene la apariencia siguiente:

```

Ejemplo_1.vb*  Ejemplo_1.vb [Diseño]*
(Ejemplo_1 eventos) Load
Public Class Ejemplo_1
    Private Sub Ejemplo_1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load
    End Sub
End Class
    
```

Como se ve la estructura del programa para una aplicación para Windows tiene como principal lo siguiente:

```
Public Class Nombre_Formulario
    Private Sub Nombre_Evento ( ... ) Tipo
        ....
    End
Sub
    ....
End
Class
```

La estructura del código se mantiene con respecto al **Visual Basic 6.0** por el hecho de trabajar con **Eventos**; más sin embargo, todos estos eventos se escribirán para la clase del formulario que lo contiene. Esto fundamenta de que **Visual Basic .NET 2010** es un lenguaje que pertenece a la **Programación Orientado a Objeto**.

3.3- Metodología de Solución

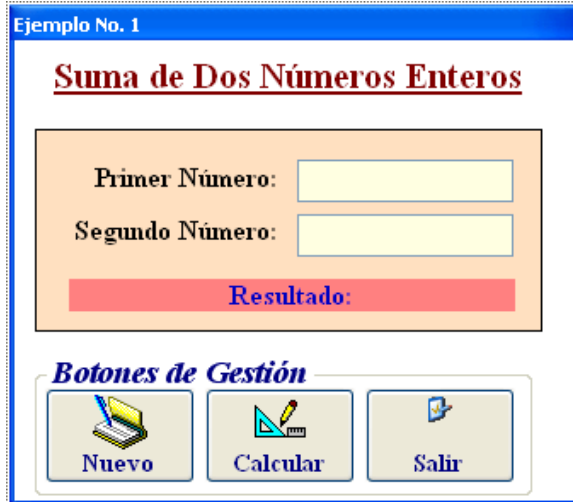
Como ya hemos visto, para resolver un problema utilizando una **Aplicación para Windows** es necesario realizar los pasos siguientes:

- ❖ Establecer la **Configuración Inicial** de la Aplicación.
- ❖ Seleccionar el tipo de aplicación **de Aplicación para Windows**.
- ❖ Desarrollando **la Interfaz Gráfica de Usuario** en el formulario.
- ❖ Establecer las **Propiedades para cada Objeto** utilizado en la aplicación.
- ❖ **Definición del Proyecto**, es decir definir el **Nombre** del proyecto y su **Descripción** de lo que hará, así como también **Establecer** qué objeto se deberá cargar inicialmente.
- ❖ Escribir el **Código de la Aplicación** en el Editor de Código.
- ❖ **Ejecutar la Aplicación**.

Ejemplo:






1. Desarrolle una aplicación en **Visual Basic .NET 2010** que sume dos números enteros.

Además, asegúrese que la apariencia de esta aplicación sea como se presenta en la figura siguiente:



Paso 1: Creando la Interfaz (Pantalla de Usuarios)

Asumiendo que se realizó los procedimientos de **Configuración Inicial** y se halla escogido el **Tipo de Aplicación**, se procede a realizar este paso. Para crear la interfaz de la aplicación simplemente consiste en utilizar todos los objetos que se usarán en la aplicación. En nuestro ejemplo, los objetos son: **Etiquetas (Label)**, **Cuadros de Textos (TextBox)**, **Botones de Comandos (Button)**, **Cuadros de Grupos (GroupBox)** y **Panel (Panel)**. Estos objetos se encuentran en la **Barra de Herramientas Estándar**.

-  ← Etiquetas (Label)
-  ← Cuadros de Textos
- (TextBox)  ← Botón de Comando (Button)
-  ← Panel (Panel)
-  ← Cuadro de Grupos (GroupBox)

Paso 2: Estableciendo las propiedades para cada objeto

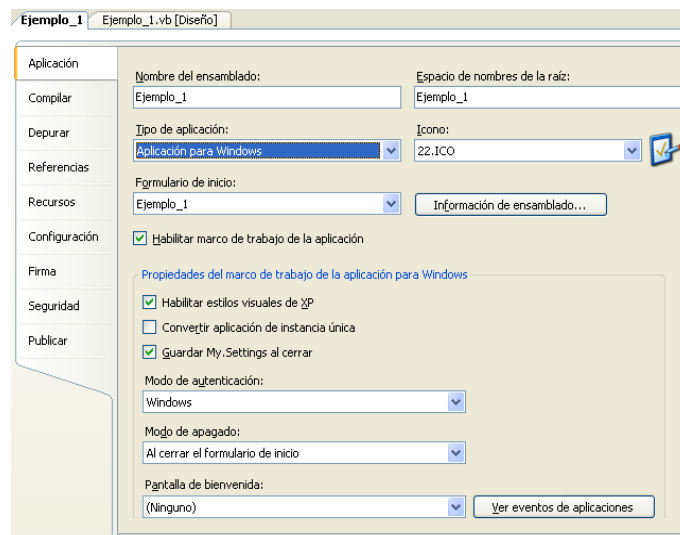
Para cada objeto seleccionado (dando clic sobre el objeto) modificamos sus propiedades mediante manipular la **Ventana de Propiedades**. En este caso, utilice dicha ventana y de los valores para cada propiedad especificada en la tabla siguiente:

| Objeto | Propiedad | Valor |
|------------------|--|---|
| Form | Text Name BackColor ControlBox FormBorderStyle StartPosition | Ejemplo No. 1 Ejemplo_1 White False FixedToolWindows CenterScreen |
| Label 1 | Text Name Font ForeColor BackColor AutoSize TextAlign | Suma de Dos Números Enteros Lbl_Titulo Times New Roman, Negrita, 16, Subrayado Rojo Maroon Transparente True MiddleCenter |
| Panel | Name BackColor BorderStyle | Fr_Datos Anaranjado FixedSingle |
| Label 2 | Text Name Font ForeColor BackColor AutoSize | Primer Número: Lbl_Primer_Numero Times New Roman, Negrita, 12 Negro Transparente True |
| Label 3 | Text Name Font ForeColor BackColor AutoSize | Segundo Número: Lbl_Segundo_Numero Times New Roman, Negrita, 12 Negro Transparente True |
| Label 4 | Text Name Font ForeColor BackColor AutoSize TextAlign BorderStyle | Resultado: Lbl_Resultado Times New Roman, Negrita, 12 Azul Rojo Claro False MiddleCenter FixedSingle |
| TextBox 1 | Name BackColor Font ForeColor MaxLength TextAlign | Txt_Num_1 Info Times New Roman, Normal, 11 Azul Oscuro 10 Center |
| TextBox 2 | Name BackColor Font ForeColor MaxLength TextAlign | Txt_Num_2 Info Times New Roman, Normal, 11 Azul Oscuro 10 Center |
| GroupBox | Name BackColor Font ForeColor Text | Fr_Gestion White Times New Roman, Negrita Cursiva, 14 Azul Oscuro Botones de Gestión |
| Button 1 | Name | Cmd_Nuevo |

| | | |
|-----------------|--|--|
| | Text BackColor Font ForeColor Image ImageAlign TextAlign | Nuevo ButtonFace Times New Roman, Negrita, 11 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |
| Button 2 | Name Text BackColor Font ForeColor Image ImageAlign TextAlign | Cmd_Calcular Calcular ButtonFace Times New Roman, Negrita, 11 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |
| Button 2 | Name Text BackColor Font ForeColor Image ImageAlign TextAlign | Cmd_Salir Salir ButtonFace Times New Roman, Negrita, 11 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |

Paso 3: Estableciendo las Propiedades del Proyecto

- ❖ Guardar los cambios realizados hasta ahora. Del **Explorador de Soluciones** seleccione el archivo **Form1.vb** de tal modo que en la **Ventana de Propiedades** en su propiedad **Nombre del Archivo** escriba **Ejemplo_1.vb**
- ❖ Guardar los cambios realizados hasta ahora. Posteriormente, de clic en la opción **Proyectos** de **Visual Basic .NET 2010**, seleccione la opción **Propiedades del Ejemplo_1** configurarlo de acuerdo a los parámetros de la ventana siguiente:



- ❖ En la misma ventana de clic en el botón **Información de Ensamblado**, configurarlo de la siguiente manera:

Paso 4: Escribiendo el Código

- ❖ Guardar los cambios realizados hasta ahora. Para acceder al Editor de Código del **Visual Basic .NET 2010** basta con dar doble clic sobre el formulario. Escriba lo siguiente:

Option Explicit On 'Asegura la Declaración de Variables
Option Strict On 'Asegura los tipos de datos asignados

```
Public Class Ejemplo_1
'Declaración de Variables
Dim Num_1 As Integer, Num_2 As Integer, Num_3 As Integer

Private Sub Ejemplo_1_Load ( ... ) ....
'Limpiando campos
Txt_Num_1.Text = " "
Txt_Num_2.Text = " "
Lbl_Resultado.Text = "Resultado: "

'Bloqueando Campos
Txt_Num_1.ReadOnly = True
Txt_Num_2.ReadOnly = True

'Botones de Comandos
Cmd_Nuevo.Enabled = True
Cmd_Calcular.Enabled = False
End Sub
End Class
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Nuevo** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Nuevo_Click ( ... ) ...  
    'Limpiando campos  
    Txt_Num_1.Text = " "  
    Txt_Num_2.Text = " "  
    Lbl_Resultado.Text = "Resultado: "  
  
    'Bloqueando Campos  
    Txt_Num_1.ReadOnly = False  
    Txt_Num_2.ReadOnly = False  
    Txt_Num_1.Focus ()  
  
    'Botones de Comandos  
    Cmd_Nuevo.Enabled = False  
    Cmd_Calcular.Enabled = True  
End Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Calcular** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Calcular_Click ( ... ) ...  
    'Verificando Campos  
    If (Txt_Num_1.Text = "") Or (Txt_Num_2.Text = "") Then  
        MessageBox.Show ("Debe de llenar todos los campos del Formulario", "Ejemplo No. 1",  
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation )  
        Exit Sub  
    End If  
  
    'Asignando Datos  
    Num_1 = CInt (Txt_Num_1.Text)  
    Num_2 = CInt (Txt_Num_2.Text)  
    Num_3 = (Num_1 + Num_2)  
  
    'Asignando Resultado  
    Lbl_Resultado.Text = "Resultado: " & Space ( 2 ) & Num_3  
  
    'Bloqueando Campos  
    Txt_Num_1.ReadOnly = True  
    Txt_Num_2.ReadOnly = True  
  
    'Botones de Comandos  
    Cmd_Nuevo.Enabled = True  
    Cmd_Calcular.Enabled = False  
End Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Salir** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Salir_Click (...) ...  
    'Fin de la Aplicacion  
End  
End Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el Textbox **Txt_Num_1** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Txt_Num_1_KeyPress (...) ...  
    'Solo Numero  
    If (e.KeyChar = Chr ( Keys.Enter )) Then  
        If ( Txt_Num_1.ReadOnly = True ) Then  
            e.Handled = True  
            Exit Sub  
        End If  
        Txt_Num_2.Focus ()  
    ElseIf (e.KeyChar <> Chr ( Keys.Back )) Then  
        If ( IsNumeric ( e.KeyChar ) = False ) Then  
            e.Handled = True  
        End If  
    End If End  
Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el Textbox **Txt_Num_2** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Txt_Num_2_KeyPress (...) ...  
    'Solo Numero  
    If (e.KeyChar = Chr ( Keys.Enter )) Then  
        If ( Txt_Num_2.ReadOnly = True ) Then  
            e.Handled = True  
            Exit Sub  
        End If  
        Cmd_Calcular.Focus ()  
    ElseIf (e.KeyChar <> Chr ( Keys.Back )) Then  
        If ( IsNumeric ( e.KeyChar ) = False ) Then  
            e.Handled = True  
        End If  
    End If End  
Sub
```

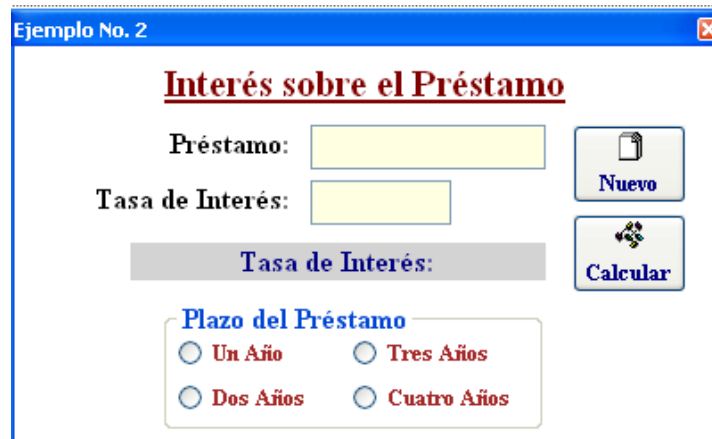
Paso 5: Compilar y Ejecutar la aplicación

Guarde los cambios realizados. Seleccione la opción **Generar** del menú principal de **Visual**

Basic .NET 2010 y escoja las opciones de **Limpiar** y **Volver a Generar**. Finalmente, seleccione la opción **Depurar** del menú principal de Visual Basic .NET 2010 y escoja la opción **Iniciar**

Depuración para que se ejecute la aplicación.

- Desarrolle una aplicación en **Visual Basic .NET 2010** que calcule el valor a pagar de un préstamo a una tasa de interés y n plazo de pago. Además, asegúrese que la apariencia de esta aplicación sea como se presenta en la figura siguiente:

**Paso 1: Creando la Interfaz (Pantalla de Usuarios)**

Asumiendo que se realizó los procedimientos de **Configuración Inicial** y se halla escogido el **Tipo de Aplicación**, se procede a realizar este paso. Para crear la interfaz de la aplicación simplemente consiste en utilizar todos los objetos que se usarán en la aplicación. En nuestro ejemplo, los objetos son: **Etiquetas (Label)**, **Cuadros de Textos (TextBox)**, **Botones de Comandos (Button)**, **Cuadros de Grupos (GroupBox)** y **Botón de Opciones (RadioButton)**. Estos objetos se encuentran en la **Barra de Herramientas Estándar**.



← Etiquetas (Label)



← Cuadros de Textos (TextBox)



← Botón de Comando (Button)



← Botón de Opciones (RadioButton)



← Cuadro de Grupos (GroupBox)

Paso 2: Estableciendo las propiedades para cada objeto

Para cada objeto seleccionado (dando clic sobre el objeto) modificamos sus propiedades mediante manipular la **Ventana de Propiedades**. En este caso, utilice dicha ventana y de los valores para cada propiedad especificada en la tabla siguiente:

| Objeto | Propiedad | Valor |
|----------------------|------------------|---|
| Form | Text | Ejemplo No. 2 |
| | Name | Ejemplo_2 |
| | BackColor | White |
| | ControlBox | True |
| | FormBorderStyle | FixedToolWindows |
| Label 1 | Text | Interés sobre el Préstamo |
| | Name | Lbl_Titulo |
| | Font | Times New Roman, Negrita, 16, Subrayado |
| | ForeColor | Rojo Maroon |
| | BackColor | Transparente |
| Label 2 | Text | Préstamo: |
| | Name | Lbl_Prestamo |
| | Font | Times New Roman, Negrita, 12 |
| | ForeColor | Black |
| | BackColor | Transparente |
| Label 3 | Text | Tasa de Interés: |
| | Name | Lbl_Tasa |
| | Font | Times New Roman, Negrita, 12 |
| | ForeColor | Black |
| | BackColor | Transparente |
| Label 4 | Text | Resultado: |
| | Name | Lbl_Resultado |
| | Font | Times New Roman, Negrita, 12 |
| | ForeColor | Azul |
| | BackColor | LightGray |
| TextBox 1 | AutoSize | False |
| | TextAlign | MiddleCenter |
| | BorderStyle | FixedSingle |
| | Name | Txt_Prestamo |
| | BackColor | Info |
| TextBox 2 | Font | Times New Roman, Normal, 11 |
| | ForeColor | Purple |
| | MaxLength | 20 |
| | TextAlign | Center |
| | Name | Txt_Tasa |
| GroupBox | BackColor | Info |
| | Font | Times New Roman, Normal, 11 |
| | ForeColor | Purple |
| | MaxLength | 10 |
| | TextAlign | Center |
| RadioButton 1 | Name | Fr_Plazo |
| | BackColor | White |
| | Font | Times New Roman, Negrita, 12 |
| | Text | Plazo del Préstamo |
| RadioButton 1 | Name | Opt_Un_Año |

| | | |
|----------------------|---|--|
| | AutoSize BackColor Font ForeColor Text | True Transparente Times New Roman, Negrita, 10 Rojo Brown Un Año |
| RadioButton 2 | Name AutoSize BackColor Font ForeColor Text | Opt_Dos_Año True Transparente Times New Roman, Negrita, 10 Rojo Brown Dos Años |
| RadioButton 3 | Name AutoSize BackColor Font ForeColor Text | Opt_Tres_Año True Transparente Times New Roman, Negrita, 10 Rojo Brown Tres Años |
| RadioButton 4 | Name AutoSize BackColor Font ForeColor Text | Opt_Cuatro_Año True Transparente Times New Roman, Negrita, 10 Rojo Brown Cuatro Años |
| Button 1 | Name Text Font ForeColor Image ImageAlign TextAlign | Cmd_Nuevo Nuevo Times New Roman, Negrita, 10 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |
| Button 2 | Name Text Font ForeColor Image ImageAlign TextAlign | Cmd_Calcular Calcular Times New Roman, Negrita, 10 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |

Paso 3: Estableciendo las Propiedades del Proyecto

- ❖ Guardar los cambios realizados hasta ahora. Del **Explorador de Soluciones** seleccione el archivo **Form1.vb** de tal modo que en la **Ventana de Propiedades** en su propiedad **Nombre del Archivo** escriba **Ejemplo_2.vb**
- ❖ Guardar los cambios realizados hasta ahora. Posteriormente, de clic en la opción **Proyectos de Visual Basic .NET 2010**, y configure los parámetros necesarios (Formulario de Inicio, Iconos) en la opción **Propiedades del Ejemplo_2**.
- ❖ En la misma ventana de clic en el botón **Información de Ensamblado**, configurar los parámetros necesarios (Título, Descripción, Compañía, Producto) para la aplicación.

Paso 4: Escribiendo el Código

- ❖ Guardar los cambios realizados hasta ahora. Para acceder al Editor de Código del **Visual Basic .NET 2010** basta con dar doble clic sobre el formulario. Escriba lo siguiente:

```
Option Explicit On           'Asegura la Declaración de Variables
Option Strict On            'Asegura los tipos de datos asignados
Imports Microsoft.VisualBasic 'Asegura las funciones de esta clase
```

```
Public Class Ejemplo_2
```

```
'Procedimiento que Inmoviliza a un formulario en un Punto de la Pantalla
```

```
Protected Overrides Sub WndProc (ByRef m As
```

```
System.Windows.Forms.
Message)
MyBase.WndProc ( m )
```

```
Const WM_MOVE As Int32 = &H3
```

```
Select Case m.Msg
```

```
Case WM_MOVE
```

```
Me.Location = New Point (280, 150)
```

```
End Select
```

```
End Sub
```

```
'Declaración de Variables
```

```
Dim Prestamo As Double, Tasa_Interes As Double, Años As Integer, Valor_Final As Double
```

```
Private Sub Ejemplo_2_Load ( ... ) ....
```

```
'Limpiando Campos
```

```
Txt_Prestamo.Text = "0.00"
```

```
Txt_Tasa.Text = "0.00"
```

```
Lbl_Resultado.Text = "Resultado: 0.00"
```

```
'Bloqueando Campos
```

```
Txt_Prestamo.ReadOnly = True
```

```
Txt_Tasa.ReadOnly = True
```

```
'Botones de Opciones
```

```
Opt_Un_Año.Checked = True
```

```
Opt_Dos_Año.Checked = False
```

```
Opt_Tres_Año.Checked = False
```

```
Opt_Cuatro_Año.Checked = False
```

```
Fr_Plazo.Enabled = False
```

```
'Botones
```

```
Cmd_Nuevo.Enabled = True
```

```
Cmd_Calcular.Enabled = False
```

```
End Sub
```

```
End Class
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Nuevo** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Nuevo_Click ( ... ) ...  
    'Limpiando Campos  
    Txt_Prestamo.Text = ""  
    Txt_Tasa.Text = ""  
    Lbl_Resultado.Text = "Resultado: "  
  
    'Bloqueando Campos  
    Txt_Prestamo.ReadOnly = False  
    Txt_Tasa.ReadOnly = False  
    Txt_Prestamo.Focus ()  
  
    'Botones de Opciones  
    Opt_Un_Año.Checked = True  
    Opt_Dos_Año.Checked = False  
    Opt_Tres_Año.Checked = False  
    Opt_Cuatro_Año.Checked = False  
    Fr_Plazo.Enabled = True  
  
    'Botones  
    Cmd_Nuevo.Enabled = False  
    Cmd_Calcular.Enabled = True  
End Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Calcular** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Calcular_Click ( ... ) ...  
    'Validando Datos  
    If (Txt_Prestamo.Text = "") Or (Txt_Tasa.Text = "") Then  
        MessageBox.Show ("Debe de llenar todos los campos del Formulario", "Ejemplo No. 2",  
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation )  
  
    Exit Sub  
End If  
  
    'Asignando Datos  
    Prestamo = CDbl ( Txt_Prestamo.Text )  
    Tasa_Interes = CDbl ( Txt_Tasa.Text )  
  
    'Procesando  
    If ( Opt_Un_Año.Checked = True ) Then
```

```

Años = 1
ElseIf ( Opt_Dos_Año.Checked = True ) Then
  Años = 2
ElseIf ( Opt_Tres_Año.Checked = True ) Then
  Años = 3
ElseIf ( Opt_Cuatro_Año.Checked = True ) Then
  Años = 4
End If
Valor_Final = Prestamo * (1 + (Tasa_Interes / 100)) ^ Años

```

```

'Asignando Valores
Lbl_Resultado.Text = "Resultado: " & Format ( Valor_Final, "###,##0.00" )

```

```

'Bloqueando Campos
Txt_Prestamo.ReadOnly = True
Txt_Tasa.ReadOnly = True
Fr_Plazo.Enabled = False

```

```

'Botones
Cmd_Nuevo.Enabled = True
Cmd_Calcular.Enabled = False
End Sub

```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el Textbox **Txt_Prestamo** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```

Private Sub Txt_Prestamo_KeyPress (...) ...
'Solo Numero
If ( e.KeyChar = Chr ( Keys.Enter ) ) Then
  If ( Txt_Prestamo.ReadOnly = True ) Then
    e.Handled = True
    Exit Sub
  End If
  Txt_Prestamo.Text = Format ( Cdbl (Txt_Prestamo.Text), "###,##0.00")
  Txt_Tasa.Focus ()
ElseIf ( e.KeyChar < > Chr ( Keys.Back ) ) Then
  If ( IsNumeric ( "0" & Txt_Prestamo.Text & e.KeyChar ) = False ) Then
    e.Handled = True
  End If
End If End
Sub

```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el Textbox **Txt_Tasa** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```

Private Sub Txt_Tasa_KeyPress (...) ...
'Solo Numero
If (e.KeyChar = Chr ( Keys.Enter ) ) Then
  If ( Txt_Tasa.ReadOnly = True ) Then
    e.Handled = True
    Exit Sub
  End If
  Txt_Tasa.Text = Format ( Cdbl (Txt_Tasa.Text), "###,##0.00")
  Cmd_Calcular.Focus ( )
ElseIf (e.KeyChar <> Chr ( Keys.Back ) ) Then
  If ( IsNumeric ( "0" & Txt_Tasa.Text & e.KeyChar ) = False ) Then
    e.Handled = True
  End If
End If End
Sub

```

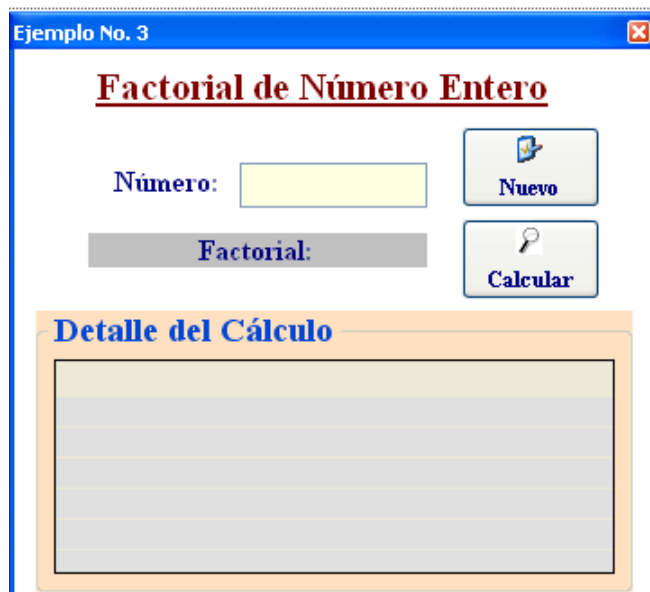
Paso 5: Compilar y Ejecutar la aplicación

Guarde los cambios realizados. Seleccione la opción **Generar** del menú principal de **Visual**

Basic .NET 2010 y escoja las opciones de **Limpiar** y **Volver a Generar**. Finalmente, seleccione la opción **Depurar** del menú principal de Visual Basic .NET 2010 y escoja la opción **Iniciar**

Depuración para que se ejecute la aplicación.

- Desarrolle una aplicación en **Visual Basic .NET 2010** que obtenga el factorial de un número entero mayor que cero. Además, que muestre un detalle desde el valor de 1 hasta ese número entero los factoriales de esos números hasta llegar al número dado. Además, asegúrese que la apariencia de esta aplicación sea como se presenta en la figura siguiente:



Paso 1: Creando la Interfaz (Pantalla de Usuarios)

Asumiendo que se realizó los procedimientos de **Configuración Inicial** y se halla escogido el **Tipo de Aplicación**, se procede a realizar este paso. Para crear la interfaz de la aplicación simplemente consiste en utilizar todos los objetos que se usarán en la aplicación. En nuestro ejemplo, los objetos son: **Etiquetas (Label)**, **Cuadros de Textos (TextBox)**, **Botones de Comandos (Button)**, **Cuadros de Grupos (GroupBox)**, **Listado de Imágenes (ImageList)** y **Listado de Vistas (ListView)**. Estos objetos se encuentran en la **Barra de Herramientas Estándar**.

Paso 2: Estableciendo las propiedades para cada objeto

Para cada objeto seleccionado (dando clic sobre el objeto) modificamos sus propiedades mediante manipular la **Ventana de Propiedades**. En este caso, utilice dicha ventana y de los valores para cada propiedad especificada en la tabla siguiente:

| Objeto | Propiedad | Valor |
|--------------------|------------------|--|
| Form | Text | Ejemplo No. 2 |
| | Name | Ejemplo_2 |
| | BackColor | White |
| | ControlBox | True |
| | FormBorderStyle | FixedToolWindows |
| Label 1 | Text | Factorial de un Número Entero |
| | Name | Lbl_Titulo |
| | Font | Times New Roman, Negrita, 16, Subrayado |
| | ForeColor | Rojo Maroon |
| | BackColor | Transparente |
| AutoSize | True | |
| Label 2 | Text | Número: |
| | Name | Lbl_Numero |
| | Font | Times New Roman, Negrita, 12 |
| | ForeColor | Black |
| | BackColor | Transparente |
| AutoSize | True | |
| Label 3 | Text | Factorial: |
| | Name | Lbl_Resultado |
| | Font | Times New Roman, Negrita, 12 |
| | ForeColor | Azul |
| | BackColor | LightGray |
| | AutoSize | False |
| | TextAlign | MiddleCenter |
| BorderStyle | FixedSingle | |
| TextBox 1 | Name | Txt_Numero |
| | BackColor | Info |
| | Font | Times New Roman, Normal, 11 |
| | ForeColor | Purple |
| | MaxLength | 10 |
| TextAlign | Center | |
| ImageList 1 | Name | Imagenes |
| | Images | Abre el Asistente e insertar 10 Imagenes |
| GroupBox | Name | Fr_Detalle |
| | BackColor | Anaranjado |
| | Font | Times New Roman, Negrita, 12 |
| | Text | Detalle del Cálculo |

| | | |
|-------------------|--|---|
| ListView 1 | Name BackColor BorderSyle Font ForeColor FullRowSelect GridLines HeaderSyle HideSelection LargeImageList MultiSelect SmallImageList View | Lv_Detalle LightGray FixedSingle Times New Roman, Negrita, 10 Azul True True NonClickAble False Imágenes False Imágenes Details |
| Button 1 | Name Text Font ForeColor Image ImageAlign TextAlign | Cmd_Nuevo Nuevo Times New Roman, Negrita, 10 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |
| Button 2 | Name Text Font ForeColor Image ImageAlign TextAlign | Cmd_Calcular Calcular Times New Roman, Negrita, 10 Azul Oscuro Seleccione una imagen que más le guste TopCenter BottomCenter |

Paso 3: Estableciendo las Propiedades del Proyecto

- ❖ Guardar los cambios realizados hasta ahora. Del **Explorador de Soluciones** seleccione el archivo **Form1.vb** de tal modo que en la **Ventana de Propiedades** en su propiedad **Nombre del Archivo** escriba **Ejemplo_3.vb**
- ❖ Guardar los cambios realizados hasta ahora. Posteriormente, de clic en la opción **Proyectos de Visual Basic .NET 2010**, y configure los parámetros necesarios (Formulario de Inicio, Iconos) en la opción **Propiedades del Ejemplo_3**.
- ❖ En la misma ventana de clic en el botón **Información de Ensamblado**, configurar los parámetros necesarios (Título, Descripción, Compañía, Producto) para la aplicación.

Paso 4: Escribiendo el Código

- ❖ Guardar los cambios realizados hasta ahora. Para acceder al Editor de Código del **Visual Basic .NET 2010** basta con dar doble clic sobre el formulario. Escriba lo siguiente:

```
Option Explicit On          'Asegura la Declaración de Variables
Option Strict On           'Asegura los tipos de datos asignados
Imports Microsoft.VisualBasic 'Asegura las funciones de esta clase
```

```
Public Class Ejemplo_3
    'Procedimiento que Inmoviliza a un formulario en un Punto de la Pantalla
```

```

Protected Overrides Sub WndProc (ByRef m As System.Windows.Forms.Message)
    MyBase.WndProc ( m )

    Const WM_MOVE As Int32 = &H3
    Select Case m.Msg
        Case WM_MOVE
            Me.Location = New Point (280, 150)
    End Select
End Sub

```

Declarando Variables

```

Dim I As Integer, Numero As Integer, Factorial As Integer
Dim Columna As ColumnHeader
Dim Fila As ListViewItem

```

```

Private Sub Encabezado ()
    'Encabezado en el Listview
    Columna = New ColumnHeader
    Columna.Text = "No"
    Lv_Detalle.Columns.Add ( Columna.Text, 50, HorizontalAlignment.Center )
    Columna.Text = "Número"
    Lv_Detalle.Columns.Add ( Columna.Text, 100, HorizontalAlignment.Center )
    Columna.Text = "Factorial"
    Lv_Detalle.Columns.Add ( Columna.Text, 150, HorizontalAlignment.Center )
End Sub

```

```

Private Sub Llenar_Lista (ByVal Col_1 As String, ByVal Col_2 As String, ByVal
Col_3

```

As String)

```

    'Llenando Listview
    Fila = New ListViewItem ( Col_1, 0 )
    Fila.SubItems.Add ( Col_2 )
    Fila.SubItems.Add ( Col_3 )
    Lv_Detalle.Items.Add ( Fila )
End Sub

```

```

Private Sub Ejemplo_3_Load ( ... ) ....

```

```

    'Limpiando Campos
    Txt_Numero.Text = "0"
    Lbl_Resultado.Text = "Factorial: 0"
    Lv_Detalle.Refresh ()
    Lv_Detalle.Clear ()

```

```

    Call Encabezado ()

```

```

    'Bloqueando Campos
    Txt_Numero.ReadOnly = True
    Fr_Detalle.Enabled = True

```

```
'Botones
  Cmd_Nuevo.Enabled = True
  Cmd_Calcular.Enabled = False
End Sub
End Class
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Nuevo** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Nuevo_Click ( ... ) ...
'Limpiando Campos
  Txt_Numero.Text = ""
  Lbl_Resultado.Text = "Factorial: "
  Lv_Detalle.Refresh ( )
  Lv_Detalle.Items.Clear ( )

'Bloqueando Campos
  Txt_Numero.ReadOnly = False
  Txt_Numero.Focus ( )

'Botones
  Cmd_Nuevo.Enabled = False
  Cmd_Calcular.Enabled = True
End Sub
```

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el botón **Calcular** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

```
Private Sub Cmd_Calcular_Click ( ... ) ...
'Verificando Campos
If (Txt_Numero.Text = "") Then
  MessageBox.Show ("Debe de Digitar el Número", "Ejemplo No. 3",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
  Exit Sub
End If

'Asignando Datos
  Numero = CInt ( Txt_Numero.Text )
If (Numero < 0) Then
  MessageBox.Show ("El Número debe ser mayor que Cero", "Ejemplo No. 3",
    MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
  Txt_Numero.Text = ""
  Txt_Numero.Focus ( )
  Exit Sub
```

End If

If (Numero = 0) **Then**

Lbl_Resultado.**Text** = "Factorial: 1"

I = 1

Factorial = 1

Call Llenar_Lista (**Str**(1), **Str**(Numero), **Str**(Factorial))

Lv_Detalle.**Ítems** (0).**Selected** = **True**

Else

Factorial = 1

I = 1

Do While (I <= Numero)

Factorial = (Factorial * I)

Call Llenar_Lista (**Str**(I), **Str**(Numero), **Str**(Factorial))

I = (I + 1)

Loop

Lbl_Resultado.**Text** = "Factorial: " & Factorial

Lv_Detalle.**Ítems** (0).**Selected** = **True**

End If

'Bloqueando Campos

Txt_Numero.**ReadOnly** = **True**

'Botones

Cmd_Nuevo.**Enabled** = **True**

Cmd_Calcular.**Enabled** = **False**

End Sub

- ❖ Guardar los cambios realizados. Cierre el **Editor de Código** y ahora seleccione el Textbox **Txt_Prestamo** y de dos veces clic para volver acceder al **Editor de Código** y escriba dentro del Segmento del **Sub ... End Sub**:

Private Sub Txt_Numero_**KeyPress** (...) ...

'Solo Numero

If (e.**KeyChar** = **Chr** (**Keys.Enter**)) **Then**

If (Txt_Numero.**ReadOnly** = **True**) **Then**

e.Handled = **True**

Exit Sub

End If

Cmd_Calcular.**Focus** ()

ElseIf (e.**KeyChar** <> **Chr** (**Keys.Back**)) **Then**

If (**IsNumeric** (Txt_Numero.**Text** & e.**KeyChar**) = **False**) **Then**

e.Handled = **True**

End If

End If End

Sub

Paso 5: Compilar y Ejecutar la aplicación

Guarde los cambios realizados. Seleccione la opción **Generar** del menú principal de **Visual**

Basic .NET 2010 y escoja las opciones de **Limpiar** y **Volver a Generar**. Finalmente, seleccione la opción **Depurar** del menú principal de Visual Basic .NET 2010 y escoja la opción **Iniciar**

Depuración para que se ejecute la aplicación.