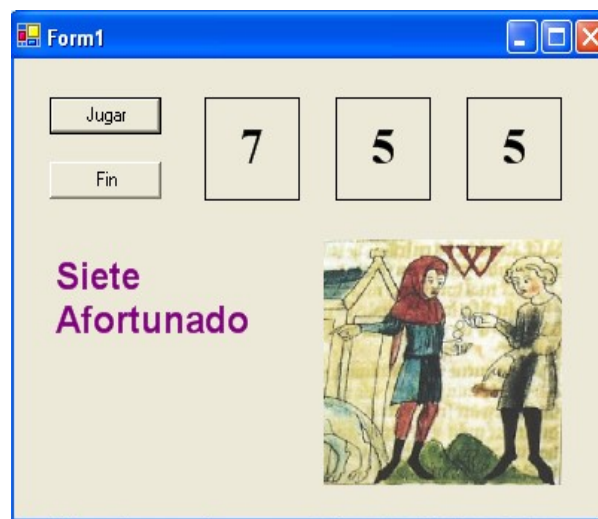


# Desarrollo del primer programa.

## 1.- Introducción.

El programa a desarrollar simula una máquina tragaperras. El aspecto del programa será:

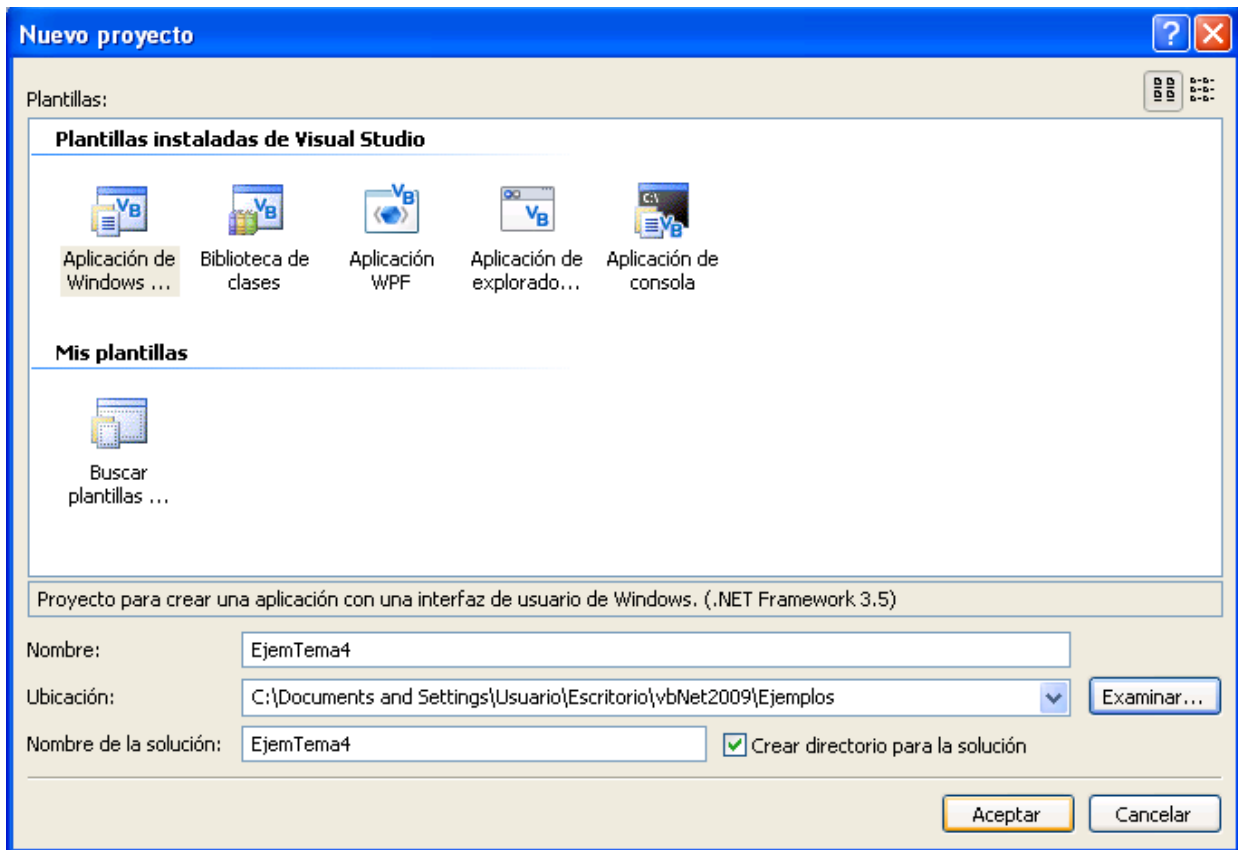


Los pasos de Programación serán:

Pasos de Programación	Número de elementos
1.-Crear la interfaz de usuario	7 objetos
2.-Definir las propiedades.	12 propiedades
3.-Escribir el código del programa.	2 objetos

## 2.- Crear un nuevo Proyecto.

Pulsar dentro de la **página de Inicio** la opción **Crear proyecto** y se mostrará un cuadro de diálogo para seleccionar el tipo de aplicación que queremos obtener.



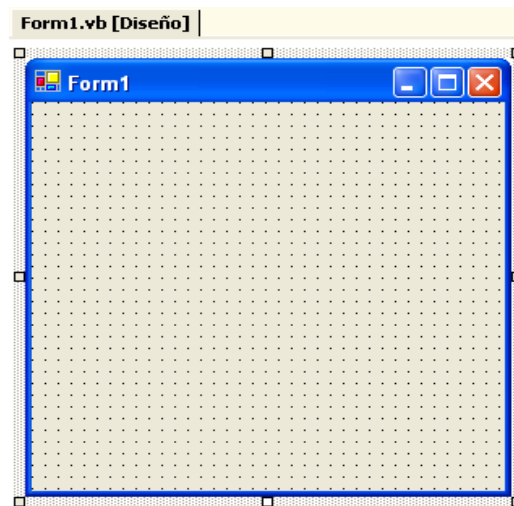
- a. Como podemos observar, en este cuadro de diálogo hay varios puntos a tener en cuenta que describimos seguidamente. También para acceder a este cuadro de diálogo se hará clic en el menú **Archivo**, y finalmente, hacer clic en **Nuevo + Proyecto**.
- b. Elegir el tipo de aplicación en la lista **Plantillas**. Seleccionaremos **Aplicación para Windows** ya que vamos a crear un programa con interfaz típica de Windows.

La gran ventaja de las plantillas radica en que al crear la aplicación, nos proporciona la funcionalidad básica de la misma, que de otro modo tendríamos que codificar manualmente.

- c. Por último, en el campo **Nombre** escribiremos **EjemTema4** como nombre para nuestra aplicación y en el campo **Ubicación** estableceremos la carpeta del disco duro que contendrá los ficheros del proyecto. Pulsando **Aceptar** se creará el nuevo proyecto.
- d. Por defecto el Nombre de la solución es igual que el nombre del proyecto, si se desea diferente nombre se puede modificar.

### 3.- Formularios.

- a. Una vez creado el proyecto, se añade un formulario al mismo, apareciendo una nueva pestaña en el área principal del IDE, que corresponde al diseñador del formulario.



- b. Dentro de una aplicación VB.NET, el término **formulario** designa a una ventana estándar de las que utilizamos habitualmente en Windows para comunicarnos con el usuario, mientras que el diseñador del formulario representa a la plantilla de una ventana, sobre la cuál añadiremos controles y modificaremos si es necesario su aspecto inicial.

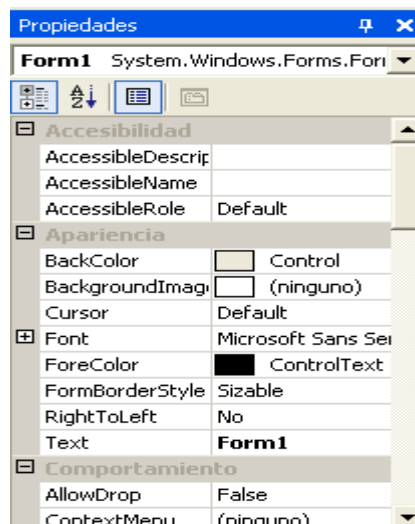
Un formulario es, al igual que la gran mayoría de elementos en el entorno de .NET, un objeto, y como tal, la forma de manipularlo pasa por asignar y obtener valores de sus propiedades, y por la ejecución de sus métodos.

- c. Debido a que un formulario dispone de un elevado número de propiedades y métodos, durante el texto nos centraremos sólo sobre los que vayamos a trabajar.

## 4.- Acceso a las propiedades de un formulario.

Para acceder a las propiedades de un formulario, podemos hacerlo de una de las siguientes maneras:

1. Seleccionar la opción **Ver + Ventana Propiedades** del menú de VS.NET.
2. Pulsar [F4].
3. Hacer clic en el botón de la barra de herramientas correspondiente a la ventana de propiedades.



En esta ventana, los nombres de las propiedades aparecen en la columna izquierda y su valor en la derecha. Por defecto, las propiedades aparecen ordenadas por categorías, pero para acceder más rápidamente a ellas a través del nombre, podemos ordenarlas alfabéticamente, pulsando el segundo botón de esta ventana comenzando por la izquierda.

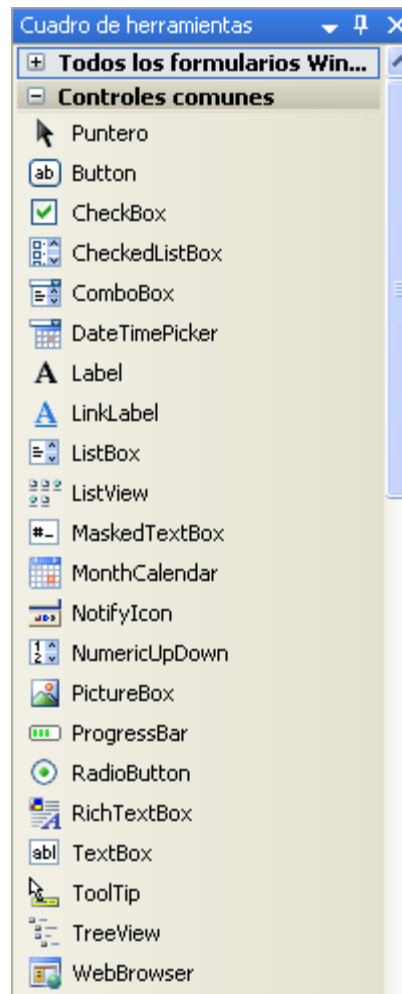


- a. La primera propiedad a mencionar, y la más importante para cualquier objeto es **Name**, que contiene el nombre del objeto que luego nos va a permitir manipularlo en el código del programa. VB.NET asigna nombres por defecto a los formularios y controles que agreguemos a la aplicación. En este caso, el nombre que se le ha asignado al formulario es **frmTragPerras**.
- b. El formulario en su estado actual, muestra como título el mismo que tiene para el nombre. La propiedad que contiene el título del formulario es **Text**, y vamos a cambiarla por un valor que describa mejor la funcionalidad que queremos dar al programa.
- c. Para ello, haremos clic sobre el valor de la propiedad Text y cambiaremos el literal que aparece por el siguiente: **TragaPerras**. Al pulsar [INTRO], el diseñador del formulario mostrará el nuevo título.
- d. Otro aspecto es referente a la posición del formulario en pantalla cuando ejecutemos el programa. Actualmente es Windows quien calcula dicha posición, apareciendo en la zona superior izquierda de la pantalla. Podemos modificar también esta posición, para ello haremos clic en la propiedad **StartPosition**, que mostrará un botón que al ser pulsado abrirá una lista con los posibles valores disponibles. Seleccionaremos **CenterScreen**, y cada vez que ejecutemos el programa, el formulario aparecerá siempre en el centro de la pantalla.

## 5- Controles.

Los controles constituyen aquellos elementos que insertamos dentro de un formulario, y que permiten al mismo interactuar con el usuario, tales como botones de pulsación, cajas de texto, casillas de verificación, cajas con listas de valores, etc.; al igual que un formulario, son objetos con sus propiedades y métodos, y se manejan de la misma forma.

- a) Para añadir un control a un formulario, en primer lugar situaremos el ratón sobre la pestaña **Cuadro de herramientas**, que al expandirse mostrará los controles disponibles, que podemos incluir en un formulario.

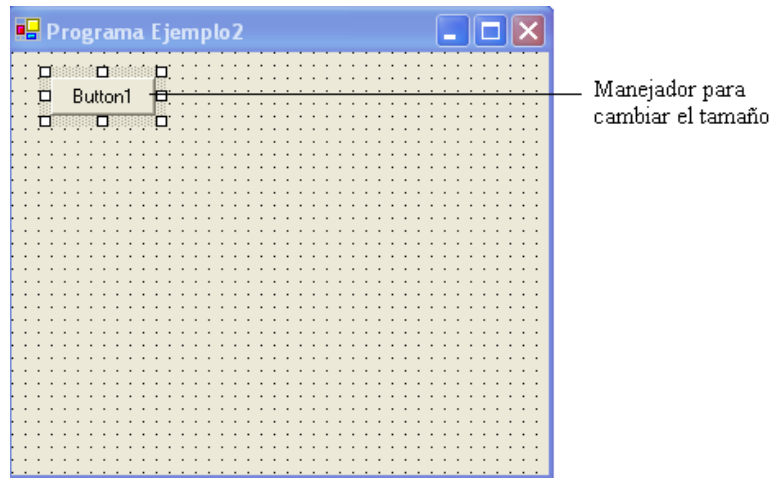


- b) La operación de añadir un control a un formulario se denomina **dibujo de control**, y podemos realizarla de varias maneras:
- Hacer doble clic sobre el icono del control, lo que situará una copia del control en el formulario con una posición y tamaño predefinidos por VB.NET.
  - Hacer clic sobre el icono del control, y a continuación clic sobre la superficie del formulario. El nuevo control se insertará desde el punto en que se ha pulsado, extendiéndose hacia la derecha y abajo con un tamaño predeterminado por el diseñador.
  - Hacer clic sobre el icono del control, y seguidamente clic sobre la posición del formulario en la que queremos comenzar a dibujar el control, arrastrar y soltar dando el tamaño que se requiera al control.

## 6.- Desarrollo del programa.

### 6. 1.- Crear la Interfaz. Agregar controles:

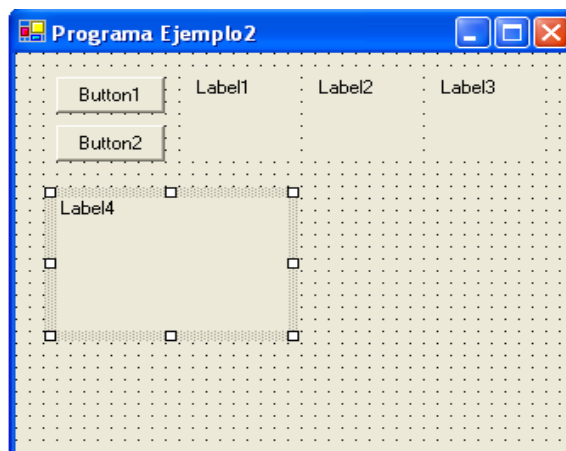
1. Hacer clic en el control **Button** contenido en el Cuadro de herramientas y desplace el puntero del ratón a una zona cercana a la esquina superior izquierda del formulario.



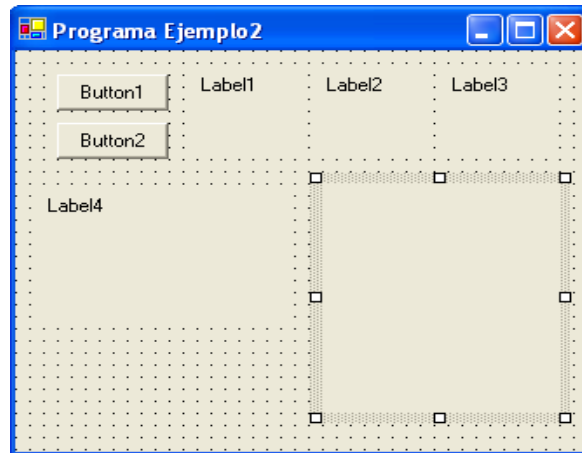
Para mover y modificar el tamaño de un control, situar el puntero del ratón sobre el control (con lo que se adoptará la forma de una cruz con cuatro puntas de flecha) y, a continuación, arrastrar el botón hacia la derecha y hacia abajo utilizando el ratón.

El botón se ajustará a la rejilla cuando suelte el ratón. El objetivo de la rejilla del formulario es ayudar a editar y alinear los diferentes elementos que aparezcan en la interfaz del usuario. Se podrá modificar el tamaño de la rejilla utilizando el comando **Opciones** contenido en el menú **Herramientas**, haciendo clic en la carpeta **Diseñador de Windows Forms** y, finalmente, modificando la propiedad **LayoutMode** con el valor **SnapToGrid**.


2. Para añadir un segundo botón repetir la acción anterior. Si se desea dibujar un control igual que otro que ya exista dibujado en el formulario, se copiará el control pulsando (**Ctrl.+C**), a continuación pulsar (**Ctrl+V**), se copiará otro control igual en tamaño y en propiedades. Mover el objeto en la posición deseada.
3. Se añadirán 4 etiquetas (Label). La Etiqueta cuarta se creará un gran rectángulo justo debajo de los dos botones. El aspecto del formulario será el siguiente:

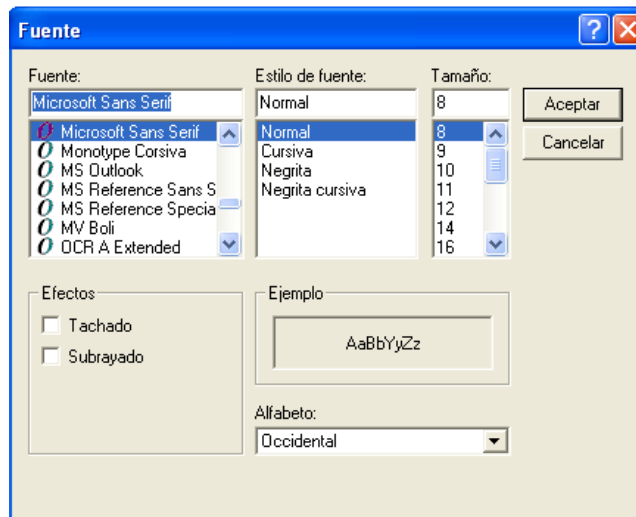


- Hacer clic en el control PictureBox , y crear un gran cuadro rectangular justo debajo de las dos etiquetas de la izquierda.



## 6. 2.- Definir las propiedades :

- Hacer clic en el primer botón (Button1), el botón quedará seleccionado y en la ventana de propiedades aparecerán las propiedades del botón seleccionado. Si la ventana de propiedades no se ve en la pantalla pulsar F4 ó .
- En la propiedad **Name** escribir **btnJugar**, en la propiedad **Text** escribir **Jugar** y pulsar INTRO. A continuación, se podrá cambiar la propiedad **Name** del segundo botón con el texto **btnFin** y la propiedad **Text** con el texto **Fin**.
- Hacer clic en la primera etiqueta numérica (label1) y mantener pulsada la tecla MAYÚS mientras hace clic con el ratón en la segunda y tercera etiquetas numéricas. El objetivo es ahora modificar el valor asignado a las propiedades **TextAlign**, **BorderStyle** y **Font**, para que los números mostrados en las etiquetas aparezcan centrados, rodeados de un recuadro y con una fuente y tamaño idénticos.
- En la propiedad **TextAlign** (Alineación del texto) desplegar el cuadro de lista desplegable, seleccionar la opción Centrado (**MiddleCenter**).
- Hacer clic con el ratón en la propiedad **BorderStyle** y, posteriormente haga clic en la flecha del cuadro de lista desplegable y seleccione **FixedSingle**, para añadir un borde fino alrededor de cada una de las etiquetas.
- Hacer clic en la propiedad **Font** en los Puntos suspensivos, en la pantalla aparecerá el cuadro de diálogo Fuente, tal y como se muestra:



7. Seleccione la fuente Times New Roman, como estilo utilice Negrita y como tamaño de punto 24. Finalmente, haga clic en Aceptar.
8. Haga clic en cualquier parte del formulario para eliminar los manejadores de selección de las tres etiquetas y, finalmente borrar de forma individual la propiedad **Text** de las etiquetas.
9. En la propiedad **Name** del objeto **Label1** se le pondrá **lblNum1**, en el objeto **Label2** se le pondrá **lblNum2** y al objeto **Label3** se le indicará **lblNum3**.
10. Modificar del objeto etiqueta **Label4** la propiedad **Text**, para introducir el rótulo "Siete Afortunado".
11. En la propiedad **Font** cambiar la Fuente a **Arial**, el estilo a **Negrita** y el tamaño de punto a **18**. Haga clic en Aceptar.
12. Ahora cambiar el color del texto, para ello hacer clic en la propiedad **ForeColor** en la ventana de Propiedades. Visual Studio muestra un cuadro de lista con tres etiquetas: Personalizado, Web y Sistema. Con las opciones contenidas en estas tres fichas se podrá definir los colores de fondo ( el color de su texto). Haga clic en la pestaña de Personalizado y seleccionar el cuadro de color morado oscuro.
13. Por último se cambiará las propiedades del control **PictureBox** que contendrá la imagen de una persona pagando una serie de monedas cuando se obtenga al menos un siete en cualquiera de las etiquetas numéricas contenidas en el formulario.
14. En la propiedad **Name** se le añadirá el texto **img**
15. Haga clic en la propiedad **SizeMode** para dar un tamaño apropiado al gráfico. Seleccione el valor **StretchImage**. Este valor hace que antes de abrir un gráfico, Visual Studio adapta el tamaño del gráfico a las dimensiones exactas del cuadro de imagen.
16. Haga clic en la propiedad **Imagen**, después haga clic en los puntos suspensivos, aparecerá el cuadro de diálogo Abrir, buscar la imagen con el nombre PayCoins.jpg y seleccionarla. La foto se cargará en el cuadro de imagen.

17. En la propiedad **Visible** seleccionar el valor **False**, para que la imagen no se muestre cuando se inicie el programa.
18. El aspecto final después de modificar las propiedades será el siguiente:



### 5. 3.- Escritura del Código:

El funcionamiento del programa vendrá dado por el estado de los botones Jugar y Fin, habrá que asociar código a dichos botones. El Editor de código es una ventana especial dentro del entorno de programación que deberá utilizar para introducir o editar sentencias de programa en Visual Basic.

1. Haga doble clic en el botón Fin del formulario. Aparecerá el Editor de código:

```
Public Class Form2
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    End Sub
End Class
```

Visual Studio añadirá automáticamente la primera y última línea del procedimiento de suceso asociado con el botón fin. También podrá observar la existencia de otras líneas de código dentro del Editor de código y que Visual Studio ha agregado para definir importantes características del formulario. Estas líneas se verán con detalle más adelante.

```
Private Sub btnFin_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles btnFin.Click
End Sub
```

El cuerpo principal de un procedimiento siempre deberá encontrarse situado entre las dos sentencias anteriores. Dichas instrucciones

se ejecutarán en cuanto el usuario active el elemento de la interfaz asociado con dicho procedimiento. En este caso, el suceso desencadenante será la pulsación del botón con el ratón.

2. Escriba **End** y pulse la tecla Flecha Abajo. Cuando escriba esta instrucción, las letras se mostrarán en azul lo que le indicará que Visual Basic la ha reconocido como una instrucción o palabra clave válida.

End es la instrucción que se utilizará para detener la ejecución del programa y eliminarlo de la pantalla.

Cuando se pulsó la tecla Flecha Abajo, se sangró la instrucción End para destacarla de las instrucciones Private Sub y End Sub.

3. Haga doble clic en el botón Jugar y se le añadirán dentro del procedimiento las siguientes líneas de código

```
Private Sub btnJugar_Click(ByVal sender As System.Object, _
    ByVal e As System.EventArgs) Handles btnJugar.Click

    ` Poner la imagen en invisible

    img.Visible = False

    ` Iniciar el generador de num. Aleatorios.
    Randomize ()
    ` Se generan 3 números aleatorios
    lblNum1.Text = Int(Rnd() * 10)
    lblNum2.Text = Int(Rnd() * 10)
    lblNum3.Text = Int(Rnd() * 10)

    `Si algún número es un 7 mostrar la imagen y emitir un sonido

    If lblNum1.Text = 7 Or lblNum2.Text = 7 _
        Or lblNum3.Text = 7 Then
        img.Visible = True
        Beep ()
    End If
End Sub
```

- La primera línea recibe el nombre de **comentario** . Los comentarios son notas explicativas incluidas en el código del programa y se caracterizan por comenzar por una comilla simple (‘). Los programadores utilizan los comentarios para describir la función que tienen en el programa las sentencias importantes.
- La instrucción **img.Visible=False** pone la imagen en invisible. Debe recordar que, anteriormente, ya definió como False el valor de esta propiedad utilizando las ventana Propiedades. Ahora, está volviendo a hacer lo mismo en el código del programa porque la primera tarea es ejecutar una nueva jugada y necesitará borrar la foto que se puede mostrar en pantalla como resultado de una jugada anterior.

- Las siguientes tres líneas son las encargadas de calcular los tres números aleatorios del 0 al 9 (ambos inclusive).


```
lblNum1.Text = Int(Rnd() * 10)
lblNum2.Text = Int(Rnd() * 10)
lblNum3.Text = Int(Rnd() * 10)
```

- El ultimo grupo de sentencias del programa comprueba si alguno de los números calculados es un siete aparecerá la imagen medieval y sonará un pitido.

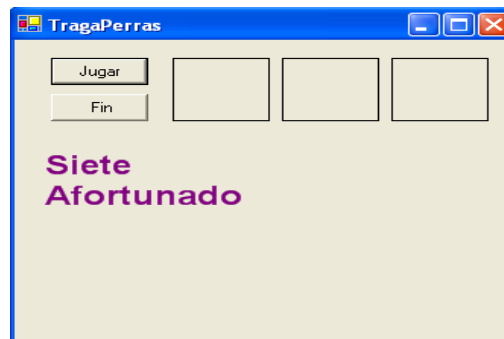
```
If lblNum1.Text = 7 Or lblNum2.Text = 7 _
Or lblNum3.Text = 7 Then
    img.Visible = True
    Beep()
End If
```

## 6.- Ejecución del programa.

En este punto del desarrollo, daremos por concluida la aplicación. Ahora debemos ejecutarla para comprobar que todo funciona correctamente. Podemos hacerlo empleando una de las siguientes formas:

1. Seleccionar la opción *Depurar + Iniciar* en el menú de VS.NET.
2. Pulsar [F5]..
3. Pulsar el botón Iniciar de la barra de herramientas. 

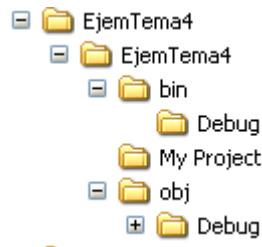
Esta acción compilará el proyecto y generará la aplicación, ejecutándola desde el propio IDE. El resultado será la visualización del formulario:



Este formulario se puede manipular de igual forma que cualquier otro de los que existan en el sistema: redimensionarlo, maximizarlo, minimizarlo, etc.

## 7.- Estructura y grabación del proyecto.

Al crear un nuevo proyecto en VB.NET, se genera en disco, partiendo de la ruta especificada en la ventana de creación del proyecto, una estructura de directorios, que contiene los archivos que forman parte del proyecto.



Si se modifica los elementos del proyecto (formularios, clases, módulos, etc), debemos grabar los cambios en alguna de las siguientes maneras:

- Opción **Generar** + **Generar** del menú de VS.NET.
- Opción **Depurar** + **Iniciar** del menú de VS.NET.
- Pulsando [F5] al ejecutar el proyecto en modo de prueba. Los elementos del proyecto que se hayan modificado y no se hayan grabado, se grabarán automáticamente.

Un proyecto VB.NET está compuesto por un conjunto de ficheros, cuyos tipos han variado notablemente desde VB6. Clasificados por su extensión, a continuación se relacionan algunos de estos ficheros:

- **VB.** Código fuente escrito en lenguaje Visual Basic. A diferencia de VB6, en el que había diferentes tipos de ficheros en función de si se trataba de un formulario, clase, módulo de código, etc., un fichero con extensión VB puede contener cualquier tipo de código en VB: clases, módulos de código, etc.
- **VBPROJ.** Proyecto de VB. Contiene información sobre todos los elementos que forman parte de un proyecto: ficheros de código, referencias, etc.
- **SLN (Solución).** Una solución es el medio que utiliza VS.NET para agrupar varios proyectos escritos en el mismo o en diferentes lenguajes de los que integran la plataforma .NET.
- **DESIGNER.vb.** contiene el diseño del formulario en código.
- **RESX.** Plantilla de recursos en formato XML.
- **EXE.** Aplicación ejecutable.
- **PDB.** Información sobre depuración de la aplicación.

## 8.- Creación de un archivo Ejecutable.

En el directorio **bin** del proyecto se genera el fichero ejecutable, que contiene nuestra aplicación y que en principio es lo único que se necesita para ejecutar el programa en cualquier otro equipo, que naturalmente, también tenga instalado la plataforma .NET Framework. Ello evita problemas y ahorra tiempo de instalación.

Visual Studio puede crear dos tipos de archivos ejecutables para el proyecto:

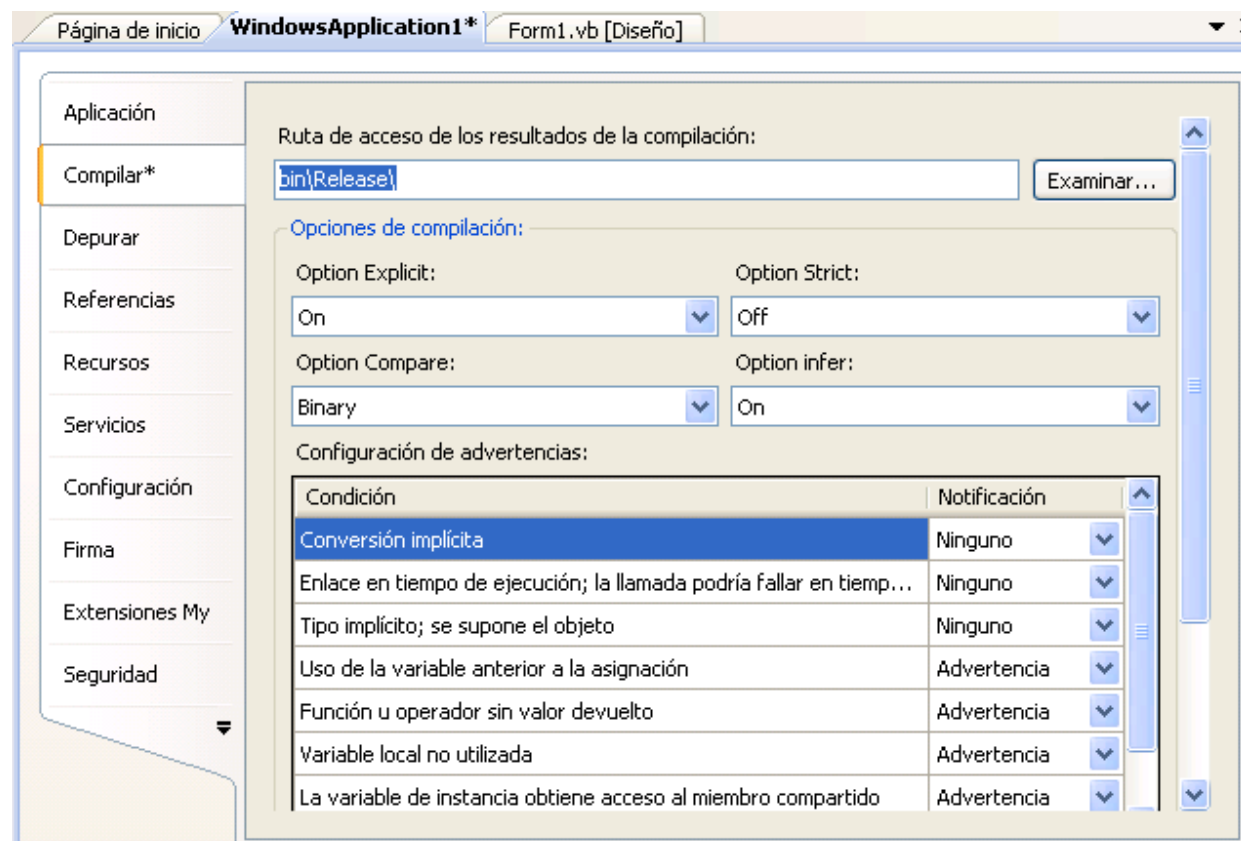
1. **Versión de compilación de depuración.** Las versiones de compilación de depuración son el tipo predeterminado en Visual Studio y las utilizará con frecuencia cuando verifique y depure sus programas. Estas versiones contienen información de depuración que hará que se ejecuten con mayor lentitud, pero el compilador de Visual Basic será capaz de producir los ejecutables con mayor rapidez. El ejecutable lo guarda en la carpeta **Debug**.
2. **Versión de edición.** Una vez terminado el desarrollo de su proyecto, se compilará con esta versión ya que incluirá numerosas optimizaciones y no contendrá información de depuración innecesaria.

Ahora se va a crear una versión de edición :

- a) En explorador de soluciones, ejecute la opción **My Project**.



Aparecerá en la pantalla de trabajo una lista de propiedades del proyecto, en ella se seleccionará **Compilar**. Donde indica **Ruta de acceso de los resultados de la compilación** se escribirá **bin\Release\**. Esto permitirá pasar de una versión de compilación de depuración a una versión de edición y viceversa. Además, este cuadro contiene otras opciones adicionales del programa. Estas opciones se verán con detalle más adelante.





- b) En el menú **Generar** ejecutar el comando **Generar solución**. El resultado es un archivo ejecutable denominado **EjemTema4.exe**.
- c) Ejecutar este programa ahora utilizando el comando Ejecutar contenido en el menú Inicio.
- d) Si volvemos a la estructura de carpetas dentro de **bin**, se habrá creado otra carpeta con el nombre **release**.