
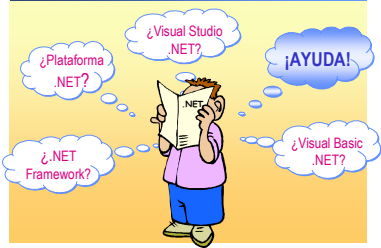
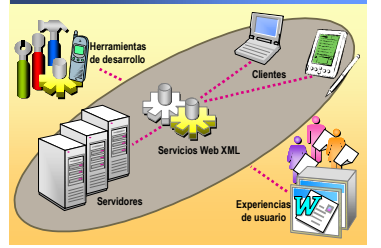
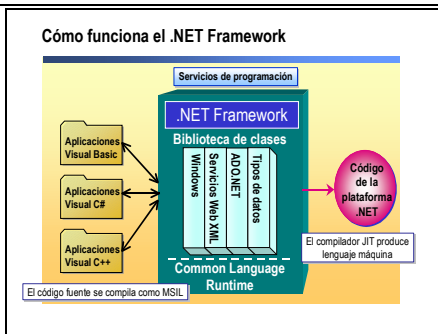




<p style="text-align: center;">Introducción a Visual Studio .NET</p>	<p>Descripción</p>  <ul style="list-style-type: none"> ■ Conceptos básicos de .NET ■ Explorar el entorno de desarrollo ■ Crear un proyecto Visual Basic .NET 												
<p>Lección: Conceptos básicos de .NET</p> 	<p>Visual Basic .NET es la última versión del sistema de desarrollo Visual Basic. Antes de empezar a crear aplicaciones en Visual Basic .NET, le será útil conocer y entender algunos conceptos básicos de .NET.</p>												
<p>¿Qué es la plataforma .NET?</p>  <p>La plataforma .NET proporciona la base de la siguiente generación de software que permite conectar sistemas, información, dispositivos y usuarios distintos de un modo más unificado y personalizado. Incorpora servicios Web XML como el medio para permitir la interoperabilidad entre tecnologías diferentes. Proporciona a los desarrolladores de software las herramientas y la tecnología para crear rápida y eficazmente soluciones de negocio que abarcan múltiples aplicaciones y múltiples dispositivos cliente entre diversas organizaciones. Permite a los usuarios controlar qué información y cómo y cuándo se les entrega.</p>	<p>La siguiente tabla proporciona definiciones y ejemplos para cada uno de los principales elementos de la plataforma .NET.</p> <table border="1" data-bbox="686 1019 1524 1926"> <thead> <tr> <th>Elemento</th> <th>Definición</th> <th>Ejemplos</th> </tr> </thead> <tbody> <tr> <td>Herramientas de desarrollo</td> <td>Interfaces y herramientas de programación para diseñar, crear, ejecutar e implantar soluciones para la plataforma .NET.</td> <td>.NET Framework Visual Studio .NET</td> </tr> <tr> <td>Servidores</td> <td>Infraestructura para generar, implantar y operar soluciones para la plataforma .NET.</td> <td>Microsoft Windows 2003 Server Servidores Microsoft.NET Enterprise Servers</td> </tr> <tr> <td>Servicios Web XML</td> <td>Conjunto centralizado de servicios predefinidos que realizan tareas habituales y rutinarias, y el medio para que los desarrolladores creen sus propios servicios.</td> <td>Servicios personalizados Microsoft .NET</td> </tr> </tbody> </table>	Elemento	Definición	Ejemplos	Herramientas de desarrollo	Interfaces y herramientas de programación para diseñar, crear, ejecutar e implantar soluciones para la plataforma .NET.	.NET Framework Visual Studio .NET	Servidores	Infraestructura para generar, implantar y operar soluciones para la plataforma .NET.	Microsoft Windows 2003 Server Servidores Microsoft.NET Enterprise Servers	Servicios Web XML	Conjunto centralizado de servicios predefinidos que realizan tareas habituales y rutinarias, y el medio para que los desarrolladores creen sus propios servicios.	Servicios personalizados Microsoft .NET
Elemento	Definición	Ejemplos											
Herramientas de desarrollo	Interfaces y herramientas de programación para diseñar, crear, ejecutar e implantar soluciones para la plataforma .NET.	.NET Framework Visual Studio .NET											
Servidores	Infraestructura para generar, implantar y operar soluciones para la plataforma .NET.	Microsoft Windows 2003 Server Servidores Microsoft.NET Enterprise Servers											
Servicios Web XML	Conjunto centralizado de servicios predefinidos que realizan tareas habituales y rutinarias, y el medio para que los desarrolladores creen sus propios servicios.	Servicios personalizados Microsoft .NET											



	Cientes	Dispositivos ejecutando sistemas operativos que se integran e interactúan con el resto de elementos de .NET.	Microsoft Windows CE para dispositivos de mano Microsoft Windows XP para PCs
	Experiencias de usuario	Software cliente convencional integrado con servicios Web XML para presentar todo lo que los usuarios necesitan de un modo que tenga sentido para ellos.	Futuras versiones de Microsoft Central™ Microsoft MSN®



El .NET Framework es un conjunto de servicios de programación diseñados para simplificar el desarrollo de aplicaciones sobre el entorno distribuido de Internet. El .NET Framework tiene dos componentes principales: el Common Language Runtime y la biblioteca de clases.

Cómo interactúan las aplicaciones con el .NET Framework

Cuando creamos una nueva aplicación Windows en Visual Basic .NET, se nos proporciona un código inicial que incluye el espacio de nombres **System.Windows.Forms** y la clase **Form**. Con esta clase, podemos crear fácilmente ventanas, botones, menús, barras de herramientas y otros elementos de pantalla. Cuando compilamos la aplicación, el código se traduce al lenguaje común del entorno de ejecución, Microsoft Intermediate Language (MSIL). Una vez la aplicación se ha compilado, el entorno de ejecución gestiona su ejecución.

El entorno de ejecución incluye una característica denominada compilación *just-in-time* (JIT), que traduce código MSIL al lenguaje máquina del sistema en el que la aplicación se ejecutará. Cuando un dispositivo cliente con la plataforma .NET lanza la aplicación en Visual Basic .NET, se ejecuta en el lenguaje máquina del sistema cliente y puede integrarse totalmente e interactuar con otras aplicaciones y servicios basados en .NET independientemente del lenguaje en el que hayan sido desarrollados.

Términos y definiciones

Para entender cómo funciona el .NET Framework, debemos estar familiarizados con la siguiente terminología:

- ❖ Una *clase* es una entidad de programación con nombre que consta de un conjunto común de métodos, propiedades y atributos. Por ejemplo, **Form** es una de las clases del espacio de nombres **System.Windows.Forms** que se utiliza para crear formularios Windows Forms.
- ❖ Espacio de nombres

Un *espacio de nombres* identifica una colección de clases relacionadas y/u otros espacios de nombres del .NET Framework. Algunos ejemplos de



espacios de nombres incluyen:

- **System**
- **System.Windows.Forms**
 - ❖ Biblioteca de clases

La *biblioteca de clases* es una colección completa orientada a objetos de clases reutilizables y organizadas en espacios de nombres jerárquicos en base a su funcionalidad. Podemos utilizar la biblioteca de clases para desarrollar aplicaciones que abarcan desde las aplicaciones cliente tradicionales hasta las aplicaciones basadas en las últimas innovaciones proporcionadas por ASP.NET y los servicios Web XML.

- ❖ Common Language Runtime

El *Common Language Runtime* es la base del .NET Framework. En el entorno .NET, los programadores desarrollan aplicaciones en el lenguaje compatible con .NET que elijan; el código se compila en MSIL, y el entorno de ejecución gestiona y ejecuta el código compilado.



Visual Studio .NET es un entorno de desarrollo integrado que nos ayuda a diseñar, desarrollar, depurar e implantar con rapidez soluciones basadas en el .NET Framework. Podemos acceder a un conjunto común de herramientas, diseñadores y editores desde cualquiera de los lenguajes de programación de Visual Studio .NET. Podemos crear aplicaciones Windows Forms y Web Forms que integren datos y lógica de negocio.

--	--

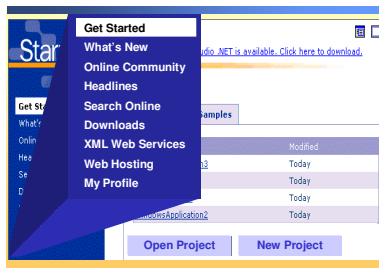
Característica	Descripción
Diseñador de Windows Forms	Una superficie de diseño gráfico que permite crear rápidamente el interfaz de usuario de una aplicación. Podemos arrastrar o dibujar controles sobre esta superficie.
Herramientas para Windows Forms	Se proporciona un Diseñador de Windows Forms, una plantilla <i>Aplicación Windows</i> , referencias de proyectos básicos y código de inicio como ayuda para crear aplicaciones Windows Forms estándares.
Herramientas para Web Forms	Se proporciona un Diseñador de Web Forms, una plantilla <i>Aplicación Web ASP.NET</i> , referencias de proyectos básicos y código de inicio como ayuda para crear aplicaciones Web Forms en



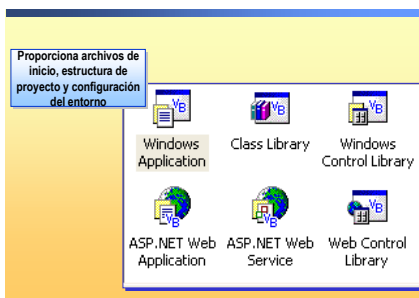
Definiciones Visual Studio .NET incluye las características de programación que se describen en la siguiente tabla		las que el interfaz de usuario principal es un navegador.
	Herramientas para servicios Web XML	Se proporciona una plantilla <i>Servicios Web ASP.NET</i> . Esta plantilla construye la estructura de un proyecto de aplicación Web en un servidor Web de desarrollo y un archivo de solución Visual Studio .NET en nuestro ordenador local.
	Soporte de múltiples lenguajes	Todos los lenguajes de programación de la plataforma .NET, incluyendo Visual Basic .NET y Visual C#, están integrados en el entorno de desarrollo.
	Acceso a datos	Componentes para crear aplicaciones que comparten datos, herramientas de bases de datos visuales para acceder a los datos y un robusto conjunto de clases de Microsoft ADO.NET facilitan el trabajo con todo tipo de datos.
	Gestión de errores	Las herramientas de depuración con soporte multilenguaje nos ayudan a encontrar y solucionar errores de código, y podemos utilizar clases de excepciones estructuradas para incluir la gestión de errores en nuestra aplicación.
	Asistentes	Los asistentes nos ayudan a completar rápidamente tareas comunes y quizás complejas. Cada página de un asistente nos ayuda a establecer opciones, configurar y personalizar proyectos.



Cómo utilizar la página de inicio



¿Qué es una plantilla de aplicación?



DEFINICIÓN: Una *plantilla de aplicación* proporciona archivos de inicio y una estructura de proyecto, y contiene los objetos básicos del proyecto y la configuración del entorno que necesitamos para crear el tipo de aplicación que deseamos.

- ❖ **Plantilla *Aplicación Windows*** Esta plantilla proporciona las herramientas, estructura y código de inicio para crear una aplicación estándar basada en Windows. Añade automáticamente las referencias básicas del proyecto y los archivos a utilizar como punto de partida para nuestra aplicación.

- ❖ **Plantilla *Aplicación Web ASP.NET***

Esta plantilla se utiliza para crear una aplicación Web ASP.NET en un equipo que tenga instalado Internet Information Services (IIS) versión 5.0 o posterior. Para ayudarnos a iniciar el diseño de nuestra aplicación, la plantilla crea los archivos básicos necesarios en el servidor.

PARA ELEGIR UNA PLANTILLA:

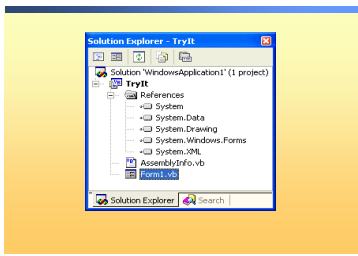
Crear un proyecto de aplicación Windows en Visual Basic .NET

1. Abra Visual Studio .NET.
2. En el panel **Inicio**, haga clic en **Nuevo Proyecto**. – O –
En el menú **Archivo**, seleccione **Nuevo**, a continuación, haga clic en **Proyecto**.
3. En el panel **Tipos de proyecto**, haga clic en **Proyectos de Visual Basic**. En el panel **Plantillas**, haga clic en **Aplicación para Windows**.
4. En el campo **Nombre**, escriba un nombre de proyecto exclusivo que indique el objetivo de la aplicación.
5. En el campo **Ubicación**, indique el directorio en el que desea guardar su proyecto, o haga clic en el botón **Examinar** para navegar hasta él.
6. Haga clic en **Aceptar**.

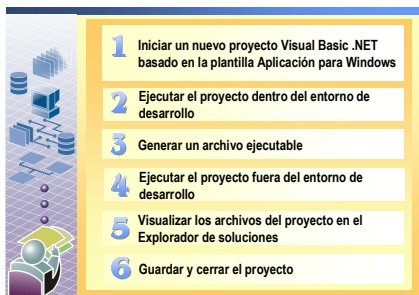
Se abrirá el Diseñador de Windows Forms, mostrando el formulario *Form1* del proyecto que ha creado.



Cómo utilizar el Explorador de soluciones



Práctica: Trabajar con un proyecto Visual Basic .NET



Iniciar el proyecto

❖ Iniciar un nuevo proyecto Visual Basic .NET

1. Abra Visual Studio .NET.
2. En el menú **Archivo**, seleccione **Nuevo** y, a continuación, haga clic en **Proyecto**.
3. En el panel **Tipos de proyecto**, haga clic en **Proyectos de Visual Basic**. En el panel **Plantillas**, haga clic en **Aplicación para Windows**.
4. En el cuadro **Nombre**, escriba **NuevoProyWindows**.
5. Haga clic en **Examinar**, navegue hasta la carpeta donde quiera crear el nuevo proyecto, y a continuación haga clic en **Abrir**. Haga clic en **Aceptar**.

Ejecutar el proyecto en el IDE

❖ Ejecutar el proyecto en el entorno de desarrollo

1. Haga clic en el botón **Iniciar** de la barra de herramientas estándar. También puede presionar F5 para ejecutar el proyecto.
Debido a que no ha realizado ningún cambio en el código de inicio para el proyecto, aparecerá un formulario en blanco cuando el proyecto se ejecute.
2. Cierre el formulario en blanco para regresar al entorno de programación.

Generar el proyecto

❖ Generar un archivo ejecutable para el proyecto

- En el menú **Generar**, haga clic en **Generar NuevoProyWindows**.
Cuando generamos un proyecto o solución, automáticamente se crea un archivo ejecutable (.exe) en la carpeta del proyecto, en una subcarpeta denominada *bin*.

Ejecutar la aplicación

❖ Ejecutar la aplicación fuera del entorno de desarrollo

- En el menú del sistema operativo **Inicio**, haga clic en **Ejecutar**, y navegue hasta el archivo .exe. El archivo .exe se encuentra en la carpeta NuevoProyWindows\bin. De la carpeta seleccionada al crear el proyecto.
Ejecutando la aplicación fuera del entorno de desarrollo de Visual Studio .NET, podemos realizar una prueba de cómo experimentará la aplicación el usuario.



Visualizar archivos

❖ Visualizar los archivos en el Explorador de soluciones

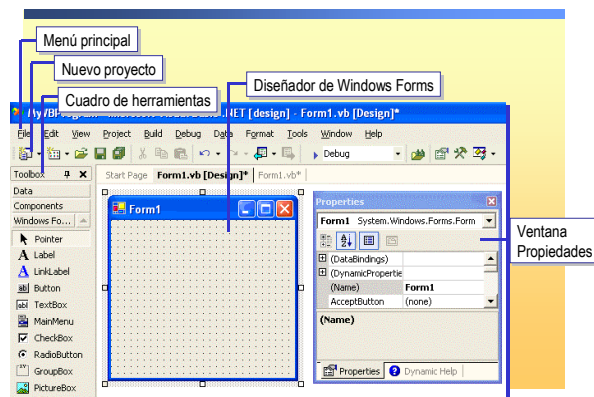
1. Si el Explorador de soluciones no está abierto, en el menú **Ver**, haga clic en **Explorador de soluciones**.
2. Expanda los nodos de archivo para visualizar los archivos. Fíjese en los archivos de solución, proyecto, AssemblyInfo.vb y Form1.vb. Estos archivos están incluidos en todos los proyectos Windows Forms.

Guardar y cerrar el proyecto

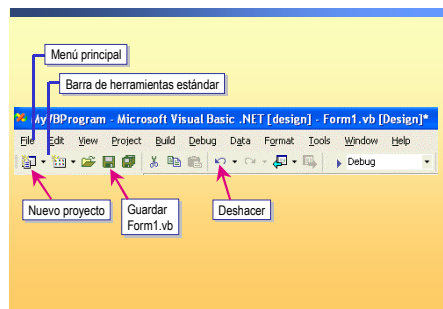
❖ Cierre el proyecto y salga de Visual Studio .NET

1. En el menú **Archivo**, haga clic en **Salir**.
2. Si se le pregunta si desea guardar los cambios, haga clic en **Sí**.

Lección: Explorar el entorno de desarrollo



Menús y barras de herramientas



Menú principal

Los comandos del menú principal funcionan igual que en todas las aplicaciones basadas en Windows, y podemos acceder a ellos utilizando el teclado o el ratón.

Barra de herramientas estándar

La barra de herramientas estándar proporciona botones que sirven como accesos directos a la mayoría de los comandos más habituales, como **Abrir**, **Guardar**, **Inicio**, **Añadir nuevo elemento**, **Ir hacia atrás** e **Ir hacia adelante**. También contiene botones que abren el Explorador de soluciones, la ventana de propiedades, el Cuadro de herramientas, y otros elementos importantes del entorno de desarrollo.

Barra de herramientas de depuración

La barra de herramientas **Depurar** proporciona accesos directos a las funcionalidades de depuración, como ir paso a paso en el código y establecer puntos de interrupción.

Funcionalidad de amarre



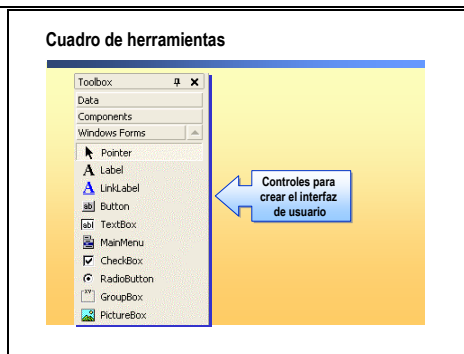
Las barras de herramientas pueden estar acopladas (adjuntas) o desacopladas (flotantes). De forma predeterminada, la mayoría de barras de herramientas están acopladas; sin embargo, podemos desacoplar o acoplar una barra de herramientas en cualquier momento.

❖ **Desacoplar una barra de herramientas**

• Haga clic en cualquier punto de la barra de herramientas salvo un botón, y arrastre la barra de herramientas fuera de su posición.

❖ **Acoplar una barra de herramientas**

• Arrastre la barra de herramientas a cualquier borde de la ventana principal.



Ocultar y reabrir el Cuadro de herramientas

❖ **Ocultar y reabrir el Cuadro de herramientas**

1. Para ocultar el Cuadro de herramientas, haga clic en el pin de la barra de título del Cuadro de herramientas.
2. Para reabrir el Cuadro de herramientas cuando está oculto, en el menú **Ver**, haga clic en **Cuadro de herramientas**.

Cambiar de lugar el Cuadro de herramientas

❖ **Cambiar de lugar el Cuadro de herramientas**

1. Haga clic con el botón derecho en la barra de título del Cuadro de herramientas y, a continuación, haga clic en **Flotante**.
2. Arrastre el Cuadro de herramientas a la posición deseada.

Introducción

Para crear los objetos del interfaz de usuario de nuestra aplicación, añadimos controles desde el Cuadro de herramientas a un formulario. Inicialmente, el Cuadro de herramientas se encuentra ubicado en el lado izquierdo del entorno de desarrollo. Hay varias pestañas para diferentes categorías de controles, como **Windows Forms** y **Datos**.

Funcionamiento del Cuadro de herramientas

El Cuadro de herramientas contiene diversos controles que podemos utilizar para añadir ilustraciones, etiquetas, botones, cuadros de lista, barras de desplazamiento, menús y formas geométricas a un interfaz de usuario. Cada control que añadimos a un formulario se convierte en un objeto de interfaz de usuario programable en nuestra aplicación. Estos objetos son visibles para los usuarios cuando la aplicación se ejecuta y funcionan como los objetos estándares de cualquier aplicación basada en Windows.

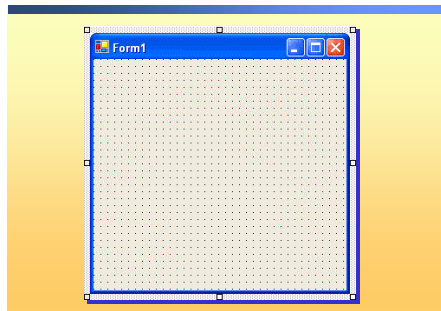
Cerrar y abrir el Cuadro de herramientas

❖ **Cerrar y abrir el Cuadro de herramientas**

1. Para cerrar el Cuadro de herramientas, haga clic en el botón **Cerrar (x)** de la esquina superior derecha del Cuadro de herramientas.
2. Para abrir el Cuadro de herramientas, en el menú **Ver**, haga clic en **Cuadro de herramientas**.
3. Para que el Cuadro de herramientas permanezca abierto, haga clic en el pin de la barra de título del Cuadro de herramientas.



Diseñador de Windows Forms



Introducción

Cuando iniciamos un proyecto en Visual Basic .NET, el Diseñador de Windows Forms se abre en la vista Diseño, mostrando el formulario *Form1* del proyecto. Ubicaremos los controles desde el Cuadro de herramientas en el formulario para crear el interfaz de usuario de una ventana utilizada en nuestra aplicación.

Elementos de un formulario

El formulario predeterminado contiene los elementos mínimos utilizados por la mayoría de formularios: una barra de título, un cuadro de control y los botones **Minimizar**, **Maximizar** y **Cerrar**.

Visualizar un formulario

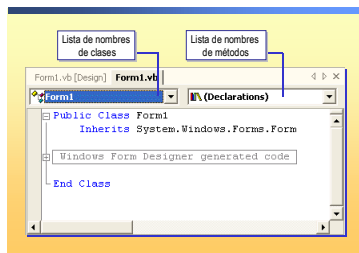
❖ Visualizar un formulario en la vista Diseño

- En el Explorador de soluciones, haga doble clic en el formulario.

– O –

En el Explorador de soluciones, seleccione el formulario y haga clic en el botón **Ver Diseñador** de la barra de herramientas.

Editor de código



Introducción

Visual Studio .NET proporciona un Editor de código en el que escribir y mantener el código de nuestro proyecto. Podemos asociar código directamente a un formulario de nuestro proyecto o ubicarlo en otro módulo de código distinto. Se muestra un Editor de código distinto para cada formulario o módulo de nuestro proyecto, facilitando la organización, visualización y desplazamiento por el código.

Funcionamiento del Editor de código

El Editor de código contiene dos listas desplegables en la parte superior de la ventana: la lista **Nombre de Clase** a la izquierda y la lista **Nombre de método** a la derecha. La lista **Nombre de Clase** muestra todos los controles del formulario asociado. Si hacemos clic en el nombre de un control de la lista, la lista **Nombre de método** muestra todos los eventos de dicho control (los eventos son acciones que el control puede realizar y que nuestra aplicación puede interpretar). Utilizando las listas **Nombre de Clase** y **Nombre de método** conjuntamente, podemos localizar rápidamente y editar el código de nuestra aplicación.

Apertura del Editor de código

❖ Abrir el Editor de código de un formulario

1. En Explorador de soluciones, haga clic en el formulario del que desea visualizar el código.

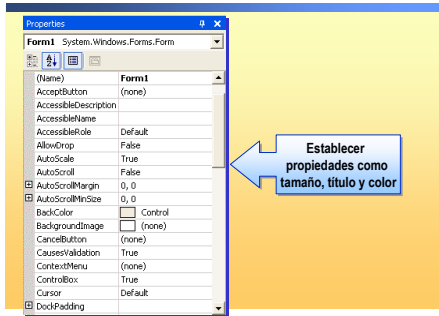
2. En la barra de herramientas de **Explorador de soluciones**, haga clic en el botón **Ver código**.

– O –

En el menú **Ver**, haga clic en **Código**.



Ventana Propiedades



Introducción

La ventana de propiedades (ventana Propiedades) muestra una lista de las propiedades que pueden configurarse para el formulario o control seleccionado y que podemos modificar mientras creamos o editamos la aplicación. Una propiedad describe una característica de un objeto, como el tamaño, título o color.

Apertura de la ventana de propiedades

❖ Abrir la ventana Propiedades

- Si la ventana Propiedades no está visible, haga clic en **Ventana Propiedades** en el menú **Ver** o pulse la tecla F4.

Ver propiedades

Algunos controles, documentos y formularios muestran un gran número de propiedades en la ventana Propiedades. Esto puede dificultar la localización de la propiedad que deseamos establecer. La ventana Propiedades nos permite visualizar las propiedades de un formulario o control en una vista ordenada por categorías en lugar de una vista alfabética.

❖ Visualizar las propiedades

1. Para visualizar las propiedades por categorías, haga clic en el botón **Por categorías** de la ventana Propiedades. Las propiedades para el formulario o control seleccionado se dividirán en categorías definidas por el control.
2. Para visualizar las propiedades de una categoría, expanda el nodo de la misma. Para ocultar las propiedades de una categoría, cierre el nodo de la misma.
3. Para visualizar las propiedades alfabéticamente, haga clic en el botón **Alfabético** de la ventana Propiedades.

Otras ventanas de programación

Ventana de programación	Objetivo
Lista de tareas	Ayuda a organizar y gestionar el trabajo de generar la aplicación
Resultados	Muestra mensajes de estado para varias características en el entorno de desarrollo
Vista de clases	Permite examinar el código tras las clases y navegar por los símbolos de la solución
Comandos	Permite emitir comandos o evaluar expresiones en el entorno de desarrollo
Examinador de objetos	Permite visualizar objetos y sus miembros

Sistema de ayuda en línea



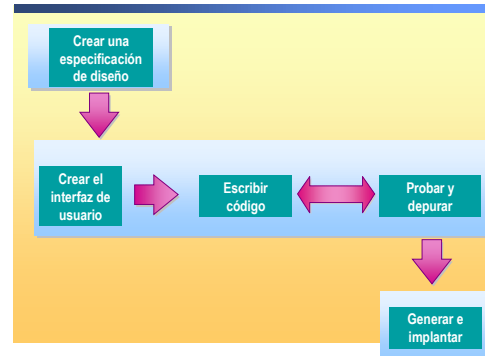


Práctica: Trabajar en el entorno de desarrollo

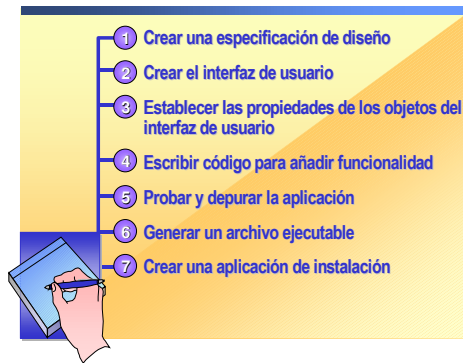


- 1 Abrir y ejecutar una aplicación existente
- 2 Examinar un formulario en el Diseñador de Windows Forms y el Editor de código
- 3 Abrir, cerrar, reabrir y ocultar el Cuadro de herramientas
- 4 Examinar la configuración de las propiedades de los controles
- 5 Utilizar la ventana Ayuda dinámica

Lección: Crear un proyecto Visual Basic .NET

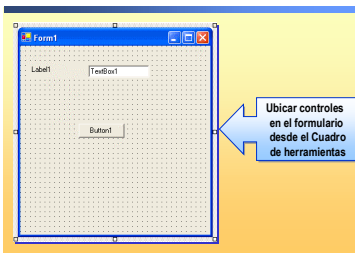


El proceso de desarrollo





Cómo crear el interfaz de usuario



Introducción

El primer paso cuando se crea una nueva aplicación es crear el interfaz de usuario. En Visual Basic .NET, podemos crear rápidamente el interfaz de usuario añadiendo controles al formulario desde el Cuadro de herramientas.

Añadir controles a un formulario

❖ Añadir controles a un formulario

1. Si el Cuadro de herramientas no está abierto, en el menú **Ver**, haga clic en **Cuadro de herramientas**.
2. En el Cuadro de herramientas, haga clic en el control que desea añadir al formulario y arrástrelo a la ubicación deseada en el formulario.

Cambiar los controles de posición y su tamaño

❖ Cambiar de posición y el tamaño de los controles

1. Para cambiar la posición del control, haga clic en el mismo para seleccionarlo, y arrástrelo a la posición deseada en el formulario.
2. Para cambiar el tamaño del control, haga clic en el control para seleccionarlo, arrastre uno de los extremos de ajuste del tamaño hasta que el control tenga el tamaño deseado y suelte el botón del ratón.

Cómo establecer las propiedades de los controles

Propiedades	Configuración
(Name)	Textbox1
BackColor	Blue
Autosize	True
Visible	True
Border	Fixed 3D
Font	Microsoft SanSerif, 8.2 pt
Text	Textbox1

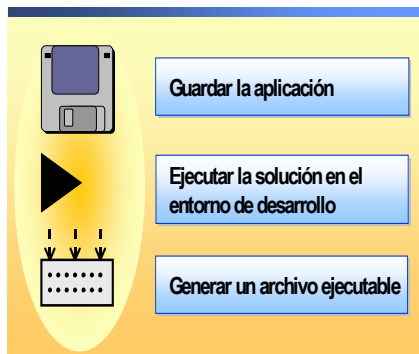
Cómo añadir código a los controles

- En la lista Nombre de clase, hacer clic en el control
- En la lista Nombre de método, hacer clic en el evento
- Añadir código entre Private Sub y End Sub

```
Private Sub Button1_Click(. . .)Handles Button1.Click  
    'Your code goes here  
End Sub
```



Como guardar, ejecutar y generar la aplicación



Introducción

Una vez hemos finalizado la creación de nuestra aplicación, la guardamos y ejecutamos dentro del entorno de desarrollo para asegurarnos de que funciona correctamente. Esta versión de la aplicación se denomina versión de depuración. A continuación, podemos generar un archivo ejecutable de la aplicación para que pueda ejecutarse fuera del entorno de desarrollo. Esta versión de la aplicación se denomina versión *release*.

Guardar la aplicación

❖ Guardar la aplicación

- En el menú **Archivo**, haga clic en **Guardar todo**.

Ejecutar el proyecto

❖ Ejecutar el proyecto dentro del entorno de desarrollo

1. En la barra de herramientas estándar, haga clic en el botón **Iniciar** para crear una versión de depuración de la aplicación.

Si la aplicación se ejecuta sin errores, aparecerá una versión ejecutable del formulario.

2. Cierre el formulario para regresar al entorno de programación.

Generar el proyecto

❖ Para generar un archivo ejecutable para el proyecto

1. En el menú **Generar**, haga clic en **Generar solución**.

Cunado generamos un proyecto o solución, automáticamente se crea un archivo ejecutable (.exe) en la carpeta *carpeta_proyecto\bin*. Si Debug está seleccionado en la lista **Configuración de la solución** de la barra de herramientas, se creará un segundo archivo .exe en la carpeta *carpeta_proyecto\obj\Debug*. Si se selecciona Release, el segundo archivo .exe se creará en la carpeta *carpeta_proyecto\obj\Release*.

2. Para ejecutar este proyecto en Windows, utilice el comando **Ejecutar** del menú **Inicio** o haga doble clic en el nombre del archivo en Windows Explorer.