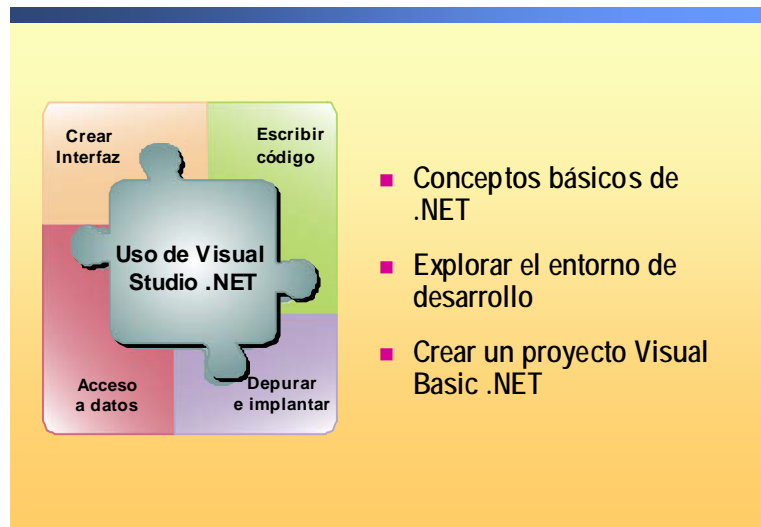


Introducción a Visual Studio .NET

Índice

Descripción	1
Lección: Conceptos básicos de .NET	2
Lección: Explorar el entorno de desarrollo	16
Lección: Crear un proyecto Visual Basic .NET	30

Descripción



Introducción

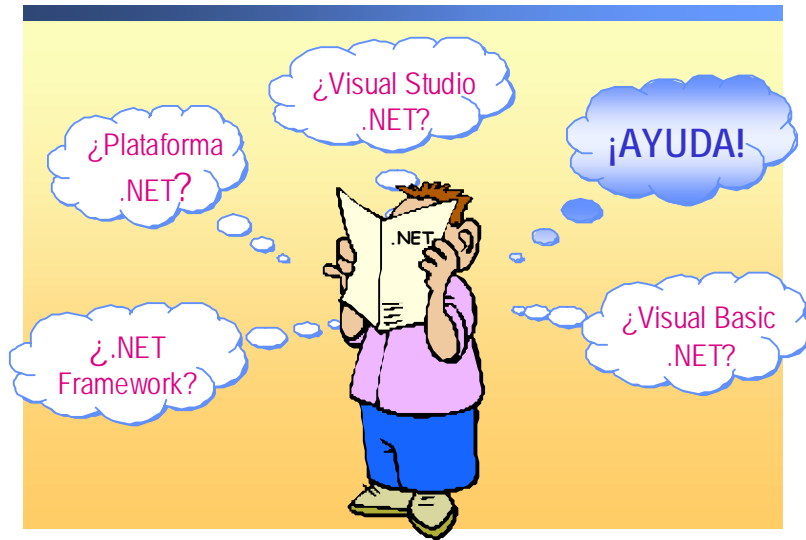
Este módulo presenta Microsoft® Visual Basic® .NET y explica cómo se integra en la plataforma Microsoft .NET. Describe cómo utilizar las herramientas de programación de Microsoft Visual Studio® .NET y proporciona suficiente actividad práctica para que podamos crear nuestra primera aplicación en Visual Basic .NET.

Objetivos

En este módulo, aprenderemos a:

- Iniciar un proyecto Visual Basic .NET basado en la plantilla *Microsoft Windows® Application*.
- Utilizar la barra de herramientas estándar, las herramientas de programación y las ventanas de programación del entorno de desarrollo para trabajar con las aplicaciones.
- Crear una aplicación sencilla en Visual Basic .NET.

Lección: Conceptos básicos de .NET



Introducción

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.

Estructura de la lección

Esta lección incluye los siguientes temas y actividades:

- ¿Qué es la plataforma .NET?
- Cómo funciona el .NET Framework
- ¿Qué es Visual Studio .NET?
- Cómo utilizar la página de inicio
- ¿Qué es una plantilla de aplicación?
- Cómo utilizar el Explorador de soluciones
- Práctica: Trabajar con un proyecto Visual Basic .NET

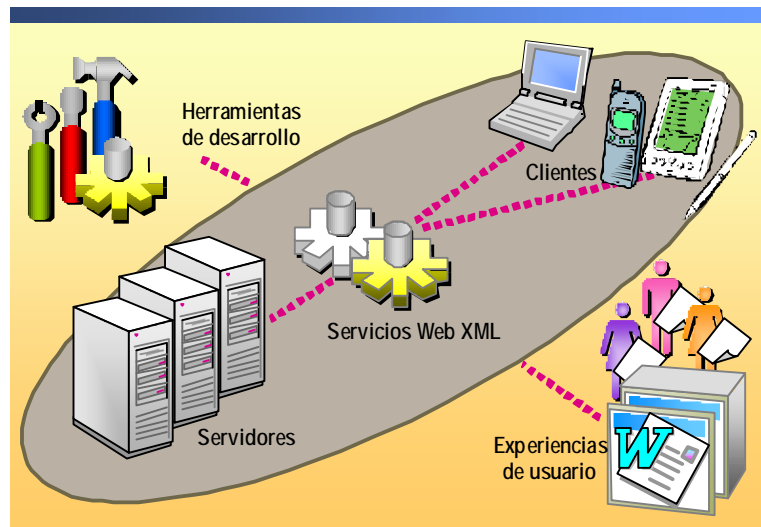
Objetivos de la lección

Esta lección ofrece una visión general de la plataforma .NET y los beneficios para los desarrolladores. Explica cómo utilizar las herramientas del entorno de desarrollo de Visual Studio .NET para crear aplicaciones en Visual Basic .NET.

En esta lección, aprenderá a:

- Describir los principales elementos de la plataforma .NET.
- Explicar los beneficios que el .NET Framework ofrece a los desarrolladores.
- Describir el papel de Visual Studio .NET y Visual Basic .NET en la plataforma .NET.
- Utilizar la página de inicio de Visual Studio para acceder a las novedades para los desarrolladores y establecer preferencias de usuario.
- Iniciar un proyecto Visual Basic .NET basado en la plantilla *Aplicación Windows*.
- Utilizar el Explorador de soluciones para examinar los archivos de solución y código.

¿Qué es la plataforma .NET?



Introducción

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.

Ejemplo práctico

El siguiente ejemplo muestra cómo la plataforma .NET puede ayudar a las empresas a integrar soluciones y compartir datos par mejorar la experiencia del usuario.

Un socio de una línea aérea se dirigió a una compañía de alquiler de coches, una de las mayores del mundo, y le ofreció la oportunidad de enlazar el sistema de reservas de la compañía con el sitio Web del socio de la línea aérea. El reto para la compañía de alquiler de coches era determinar cómo transferir fácilmente las peticiones de reserva y los datos relacionados entre esta gran línea aérea, que utilizaba sistemas UNIX, y su compañía, que utilizaba principalmente tecnologías Windows.

La compañía de alquiler de coches decidió crear su solución con tecnologías .NET y servicios Web XML. La solución permitió a la compañía crear rápidamente un sistema de traducción entre el sistema UNIX de la línea aérea y el sistema VMS de la compañía. Dos personas desarrollaron la solución en sólo dos meses (entre dos y tres veces más rápido de lo que la compañía había previsto).

Una opción posible era utilizar métodos tradicionales para crear una solución única para esta línea aérea en particular. Pero al crear el sistema con servicios Web XML, la compañía de alquiler de coches puede extender su servicio a múltiples líneas aéreas y múltiples socios. Si otros socios o clientes corporativos desean conectarse directamente con el sistema de reservas de la compañía, lo único que la compañía necesita hacer es dar el esquema XML.

Familia de productos .NET

La plataforma .NET incluye una completa familia de productos creados para trabajar con los estándares de XML e Internet. Estos productos incluyen los siguientes componentes que trabajan con soluciones basadas en XML:

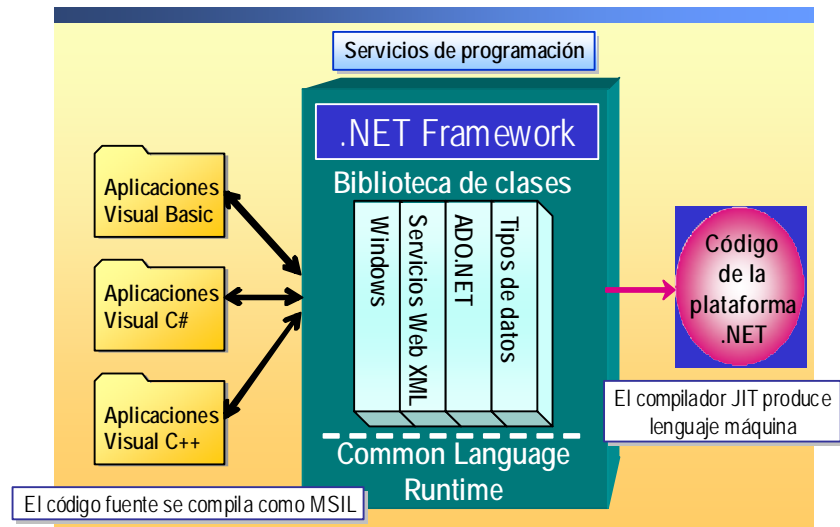
- Herramientas para desarrollar soluciones
- Servidores para gestionar, crear e implantar soluciones
- Servicios para integrar, conectar y reutilizar soluciones
- Clientes hardware y software para proporcionar experiencias sin precedentes a los usuarios

Elementos principales

La siguiente tabla proporciona definiciones y ejemplos para cada uno de los principales elementos de la plataforma .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
Clientes	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 bCentral™ Microsoft MSN®

Cómo funciona el .NET Framework



Introducción

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:

■ **Clase**

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**

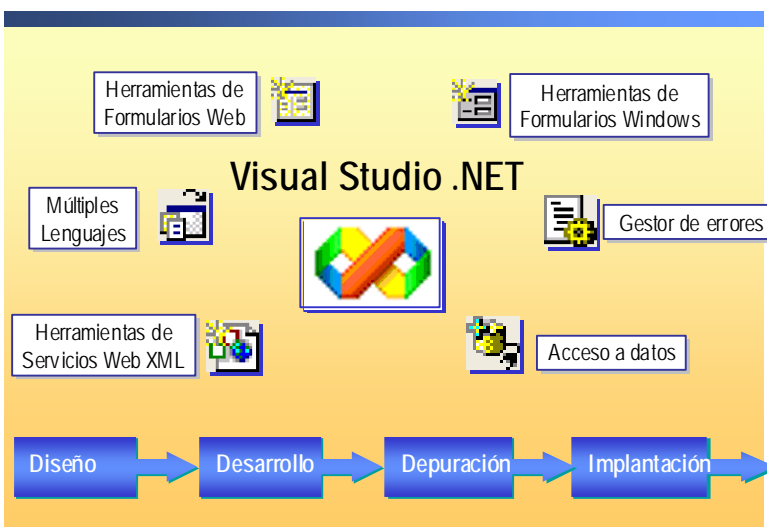
La *biblioteca de clases* es una colección completa orientada a objetos de clases reutilizables y organizadas en espacios de nombres jerárquicos en base a su funcionalidad. Podemos utilizar la biblioteca de clases para desarrollar aplicaciones que abarcan desde las aplicaciones cliente tradicionales 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.

Nota Además de los lenguajes de Microsoft como Visual Basic, Microsoft Visual C#™, Microsoft Visual C++® y Microsoft JScript®, otros lenguajes, incluyendo Perl, Smalltalk y Eiffel, han sido modificados para que sean compatibles con la plataforma .NET.

¿Qué es Visual Studio .NET?



Introducción

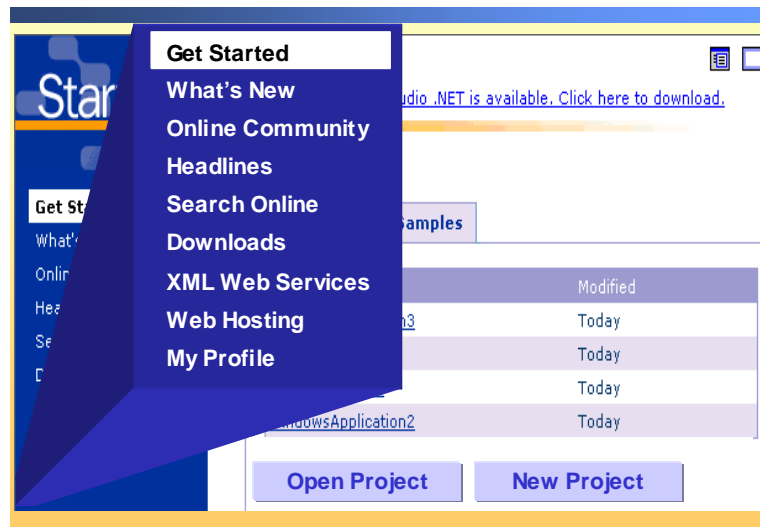
Visual Studio .NET es un entorno de desarrollo integrado que nos ayuda a diseñar, desarrollar, depurar e implantar con rapidez soluciones basadas en el .NET Framework. Podemos acceder a un conjunto común de herramientas, diseñadores y editores desde cualquiera de los lenguajes de programación de Visual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

Definiciones

Visual Studio .NET incluye las características de programación que se describen en la siguiente tabla:

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 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



Introducción

Desde la página de inicio de Visual Studio .NET, podemos abrir proyectos, crear nuestro propio perfil y establecer nuestras preferencias. También podemos leer las últimas noticias de los desarrolladores de Microsoft, participar en las comunidades online de .NET, realizar búsquedas en Microsoft MSDN® Library y descargar actualizaciones de software.

Abrir proyectos

Podemos iniciar un nuevo proyecto o abrir un proyecto existente desde la Página de inicio.

⚡ Abrir un nuevo proyecto

- Haga clic en **Inicio** en la lista del lado izquierdo de la página (si no está ya seleccionado) y, a continuación, haga clic en **Nuevo Proyecto**.

⚡ Abrir un proyecto existente

- Haga clic en **Inicio** en la lista del lado izquierdo de la página y, a continuación, haga clic en **Abrir Proyecto**.

Establecer preferencias en Visual Studio .NET

Puede escoger un perfil ya existente, como Programador de Visual Basic, o puede modificar manualmente cada elemento del perfil.

✚ Establecer las preferencias de la Página de inicio

1. En la Página de inicio, haga clic en **Mi perfil** en la lista del lado izquierdo de la página.
2. En la primera lista desplegable debajo de Perfil, haga clic en **Programador de Visual Basic** (si no está ya seleccionado).
3. Debajo de **Mostrar Ayuda**, decida si desea que el contenido de la ayuda aparezca en una ventana de navegador Web en el entorno de desarrollo o en una ventana externa.
4. En la lista **Al iniciar**, escoja qué desea que aparezca cuando inicie Visual Studio .NET.

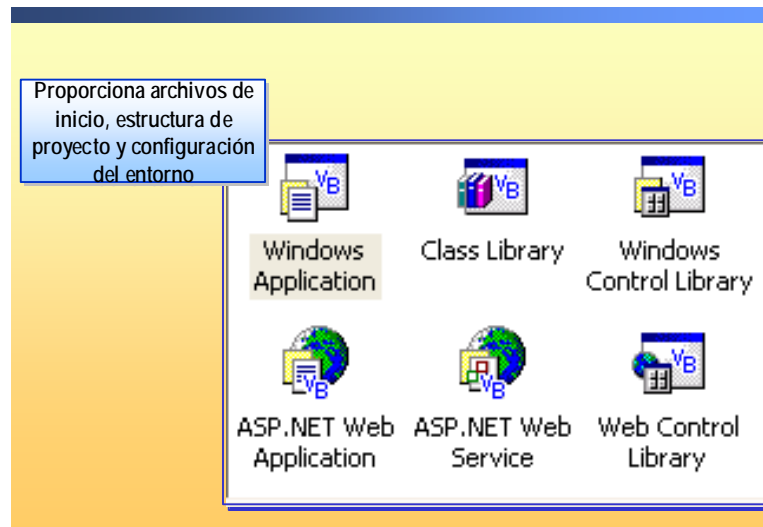
Visualizar la Página de inicio

Puede visualizar la Página de inicio en cualquier momento mientras trabaja en el entorno de desarrollo.

✚ Visualizar la Página de inicio

- En el menú **Ayuda**, haga clic en **Mostrar página de inicio**.

¿Qué es una plantilla de aplicación?



Introducción

Visual Studio .NET ofrece varias plantillas de aplicaciones para soportar el desarrollo de diferentes tipos de aplicaciones y componentes. Antes de iniciar un nuevo proyecto, debemos escoger el tipo de plantilla que deseamos utilizar.

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.

Ejemplos de plantillas de aplicaciones

Dos de las plantillas que más se utilizan habitualmente son *Aplicación Windows* y *Aplicación Web ASP.NET*.

■ 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.

Elegir una plantilla

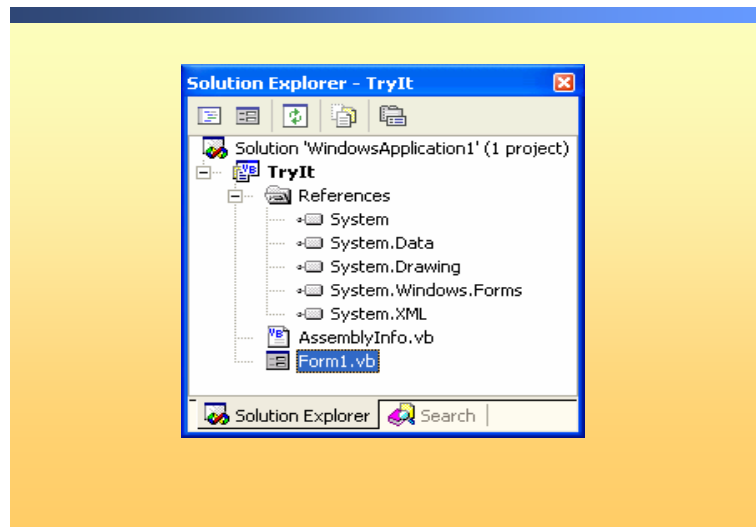
Cuando se inicia un nuevo proyecto Visual Basic .NET, uno de los primeros pasos es escoger una plantilla de aplicaciones. En este módulo, únicamente utilizaremos la plantilla *Aplicación Windows* para realizar las prácticas y los laboratorios.

✍ 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** 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 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



Introducción

El Explorador de soluciones nos permite visualizar archivos y realizar tareas de administración de archivos en una solución o en un proyecto. Una única solución basada en Visual Basic .NET y sus proyectos aparecen en una lista jerárquica que proporciona información actualizada sobre el estado de nuestra solución, proyectos y archivos.

Términos y definiciones

Una *solución* es un recipiente para proyectos y elementos de solución que pueden incluirse en una aplicación. Normalmente, una solución contiene uno o más proyectos relacionados.

Un *proyecto* es un recipiente dentro de una solución que se utiliza para administrar, generar y depurar lógicamente los elementos de proyecto que constituyen nuestra aplicación.

Archivos de proyectos y soluciones

Visual Studio .NET soporta numerosos tipos de archivos y sus extensiones de archivo asociadas. La siguiente tabla describe algunos tipos de archivos habituales específicos para las soluciones basadas en Visual Basic .NET.

Extensión	Nombre	Descripción
.sln	Solución Visual Studio	Organiza proyectos, elementos de proyectos y elementos de soluciones en una solución proporcionando al entorno referencias a sus ubicaciones en disco.
.suo	Solution user options	Registra todas las opciones que podemos asociar a nuestra solución de forma que cada vez que abramos la solución incluya las personalizaciones que hayamos realizado.
.vb	Proyecto Visual Basic	Representa los archivos de formularios, controles de usuario, clases y módulos que pertenecen a la solución de un solo proyecto. Los archivos que no estén basados en un lenguaje de programación tienen su propia extensión. Por ejemplo, un archivo Crystal Report tiene la extensión .rpt, y un archivo de texto tiene la extensión .txt.
.vbproj	Proyectos Visual Basic	Representa los archivos de formularios, controles de usuario, clases y módulos que pertenecen a la solución con múltiples proyectos. Esta extensión nos permite diferenciar entre archivos escritos en Visual Basic .NET y otros lenguajes compatibles con .NET. (Visual C# utiliza .csproj.)
.aspx .asmx .asax	Elementos de proyecto Web	Los elementos de proyecto Web incluyen archivos Web específicos como .aspx para Web Forms, .asmx para servicios Web XML, y .asax para clases globales de aplicaciones. Los proyectos Web también utilizan la extensión .vb para clases y módulos.

Abrir el Explorador de soluciones


✚ Abrir el Explorador de soluciones

- En el menú **Ver**, haga clic en **Explorador de soluciones**.
Aparecerá una ventana en la esquina superior derecha del área de desarrollo. Podemos mover y cambiar el tamaño de esta ventana mediante la funcionalidad estándar de arrastrar y soltar.

Mostrar todos los archivos

De forma predeterminada, el Explorador de soluciones únicamente muestra algunos de los archivos almacenados en el proyecto. Podemos cambiar la vista para ver todos los archivos de una solución.

✚ Mostrar todos los archivos del Explorador de soluciones

- En la barra de herramientas del **Explorador de soluciones**, haga clic en **Mostrar todos los archivos**. 
Esta opción permite ver los archivos creados manualmente o durante la compilación.

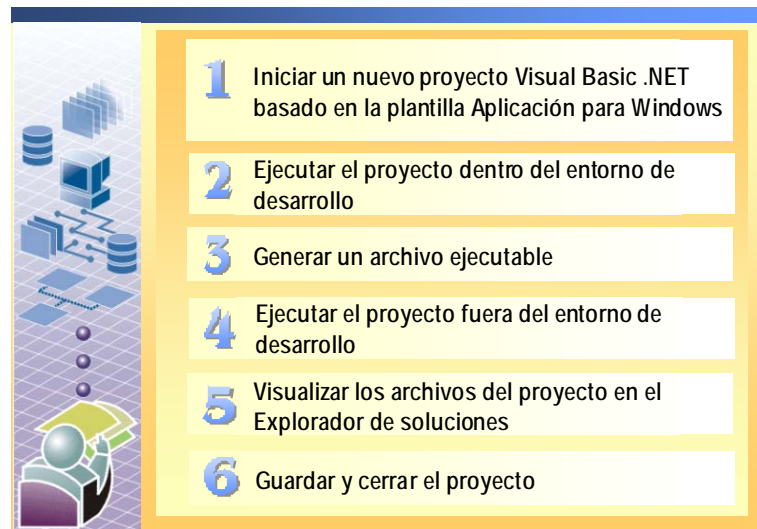
Eliminar proyectos

Podemos eliminar proyectos de forma permanente cuando ya no sean necesarios.

✚ Eliminar proyectos

- En Windows Explorer, haga clic con el botón derecho en la carpeta que contiene la solución que desea eliminar y, a continuación, haga clic en **Eliminar**.

Práctica: Trabajar con un proyecto Visual Basic .NET



Introducción

En esta práctica, iniciará un nuevo proyecto Visual Basic .NET y trabajará con él en el entorno de desarrollo.

Los archivos solución de esta practica se pueden encontrar en el fichero practs02.zip


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.

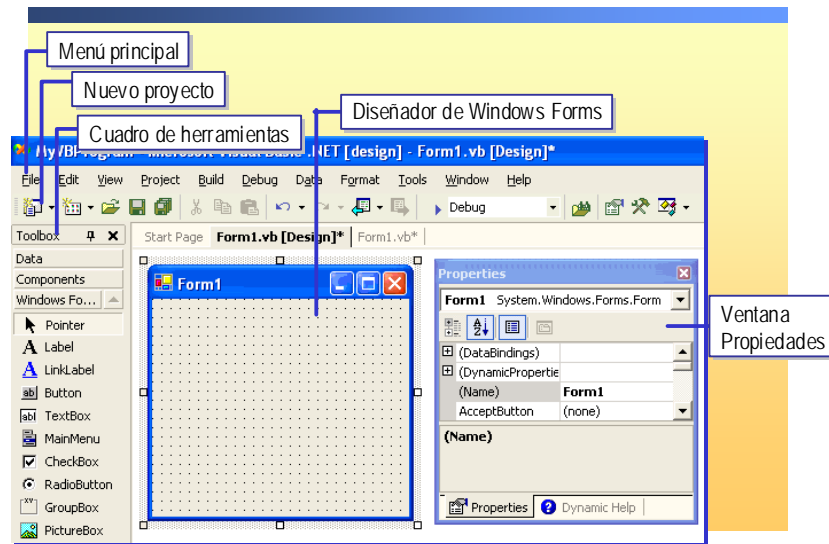
Visualizar archivos**⚡ Visualizar los archivos en el Explorador de soluciones**

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



Introducción

Esta lección introduce las principales características del entorno de desarrollo que utilizamos para crear y modificar aplicaciones, como el Diseñador Windows Forms y los controles del Cuadro de herramientas.

Estructura de la lección

Esta lección incluye los siguientes temas y actividades:

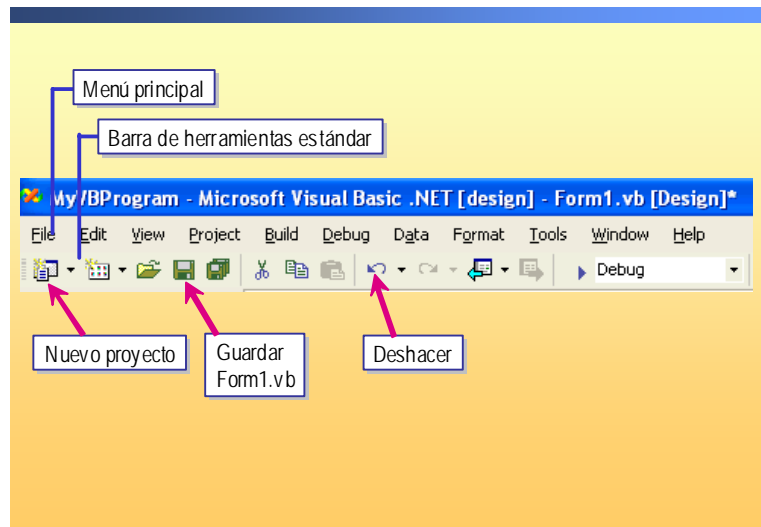
- Menús y Barras de herramientas
- El Cuadro de herramientas
- Diseñador de Windows Forms
- Editor de código
- Ventana de propiedades
- Otras ventanas de programación
- Sistema de ayuda en línea
- Práctica: Trabajar en el entorno de desarrollo

Objetivos de la lección

En esta lección, aprenderá a:

- Utilizar el menú principal y la barra de herramientas estándar para abrir, generar (modos *debug* y *release*), ejecutar, guardar y cerrar una aplicación basada en Visual Basic .NET existente.
- Visualizar los formularios del interfaz de usuario en el Diseñador de Windows Forms o en el Editor de código.
- Utilizar el Cuadro de herramientas para añadir un botón y una etiqueta a un formulario.
- Utilizar la ventana de propiedades para visualizar y configurar propiedades.
- Abrir, cerrar y volver a abrir herramientas de programación y ventanas.
- Utilizar el sistema de ayuda en línea para obtener más información.

Menús y Barras de herramientas



Introducción	Cuando iniciamos un proyecto Visual Basic .NET y seleccionamos una plantilla de aplicaciones, aparece el entorno de desarrollo visual. El menú principal y la barra de herramientas estándar proporcionan acceso a la mayoría de comandos que controlan el entorno de programación de Visual Studio .NET.
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.

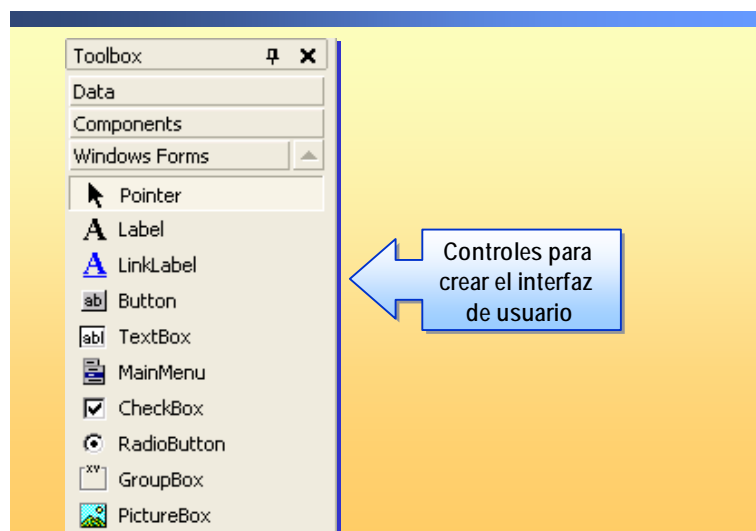
↙ Desacoplar una barra de herramientas

- 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.

Cuadro de herramientas



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.

Ocultar y reabrir el Cuadro de herramientas

🔍 Ocultar y reabrir el Cuadro de herramientas

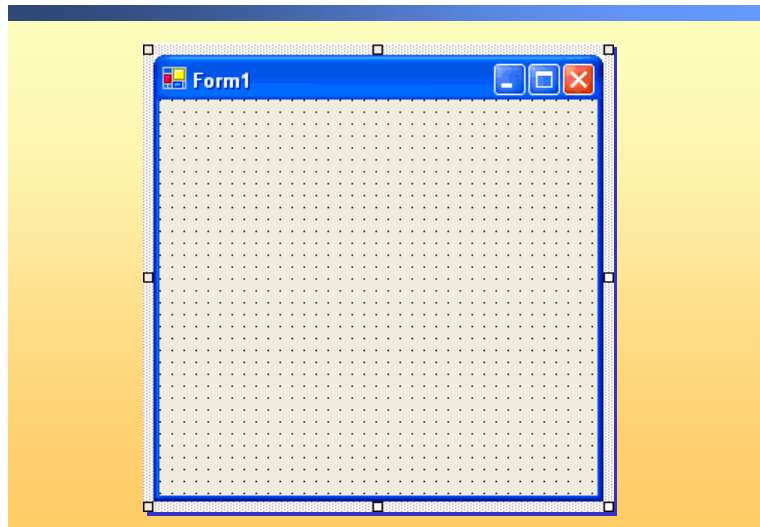
1. Para ocultar el Cuadro de herramientas, haga clic en el pin  de la barra de título del Cuadro de herramientas.
2. Para reabrir el Cuadro de herramientas cuando está oculto, en el menú **Ver**, haga clic en **Cuadro de herramientas**.

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.

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**.

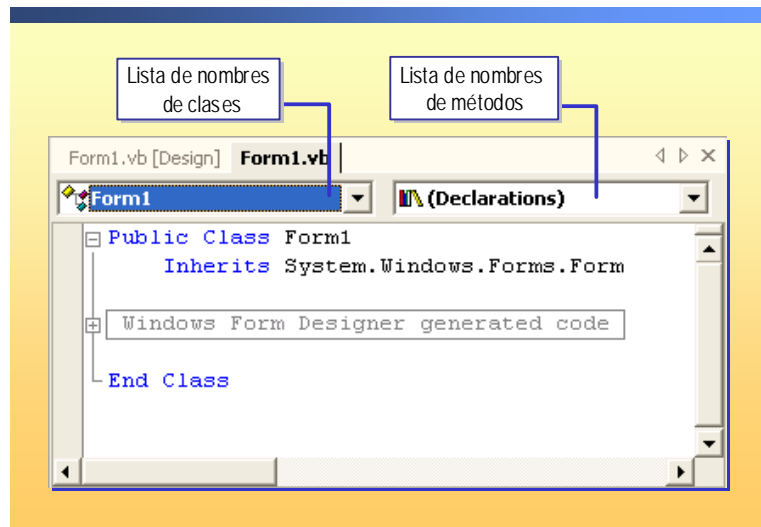
Visualizar un formulario

↙ 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

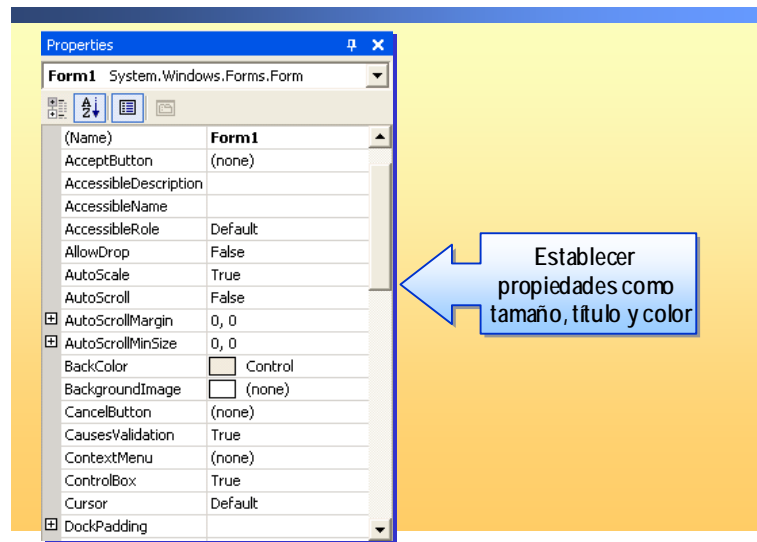
✎ Abrir el Editor de código de un formulario

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

Introducción Existen otras ventanas de programación en el entorno de desarrollo que pueden resultar útiles.

Ventana Lista de tareas La ventana Lista de tareas nos ayuda a organizar y gestionar el trabajo de generar nuestra aplicación.

⚡ Mostrar la ventana Lista de tareas

- En el menú **Ver**, seleccione **Otras ventanas** y, a continuación, haga clic en **Lista de tareas**.

Ventana Resultados La ventana Resultados muestra mensajes de estado para diversas características del entorno de desarrollo.

⚡ Acceso a la ventana Resultados

- En el menú **Ver**, seleccione **Otras ventanas** y, a continuación, haga clic en **Resultados**.

Ventana Vista de clases La Vista de clases permite examinar el código y navegar hasta los símbolos que representan elementos de programación de nuestra solución. Los símbolos, organizados por proyecto, aparecen en una vista de árbol jerárquico, indicando las relaciones de contención existentes entre ellos.

⚡ Abrir la ventana Vista de clases

- En el menú **Ver**, haga clic en **Vista de clases**.

Ventana de comandos

La ventana de comandos se utiliza para invocar comandos o para depurar y evaluar expresiones del entorno de desarrollo.

⚡ Abrir la ventana de comandos

- En el menú **Ver**, seleccione **Otras ventanas** y, a continuación, haga clic en **Ventana de comandos**.

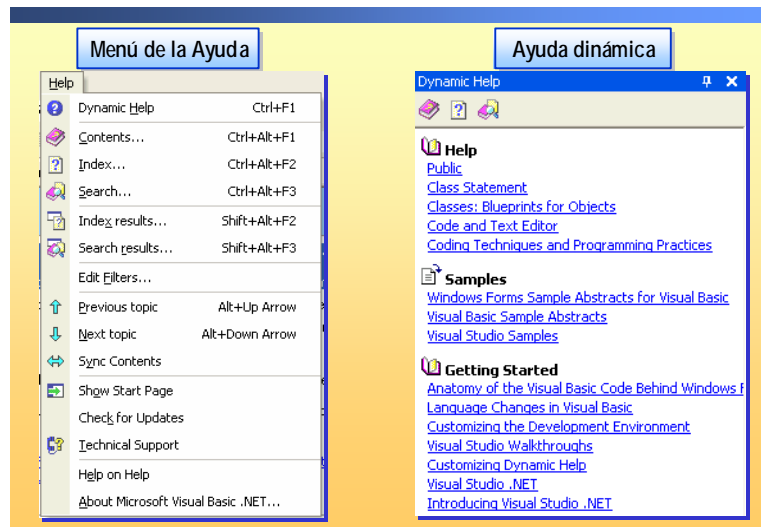
Ventana Examinador de Objetos

El Examinador de objetos permite examinar y descubrir objetos, como espacios de nombres y clases, que forman parte de nuestra solución.

⚡ Abrir el Examinador de objetos

- En el menú **Ver**, seleccione **Examinador de Objetos**.

Sistema de ayuda en línea



Introducción

La ayuda de Visual Studio .NET está completamente integrada con el entorno de desarrollo. Proporciona la información necesaria basada en la tarea que estamos realizando en ese momento. Para incrementar su eficacia al utilizar el entorno de desarrollo, piense en las situaciones en las que la Ayuda puede beneficiarle y dónde podría encontrar ayuda en cada situación.

Estrategias para utilizar la Ayuda

Hay dos formas principales de acceder a la Ayuda en el entorno de desarrollo: la ventana Ayuda dinámica y el menú **Ayuda**.

Ventana Ayuda dinámica

La ventana Ayuda dinámica proporciona punteros a información específica al área que estamos utilizando en cada momento o a la tarea que intentamos llevar a cabo en el entorno de desarrollo. El entorno de desarrollo hace un seguimiento de las selecciones que realizamos, de la ubicación del puntero y de los elementos que tienen el foco mientras interactuamos con el producto. La ventana Ayuda dinámica filtra los temas disponibles en Microsoft Developer Network (MSDN) para mostrar la información relevante. Si no puede encontrarse información relevante, la ventana muestra el texto “No hay enlaces disponibles para esta selección”).

🔍 Abrir la ventana Ayuda dinámica

- En el menú **Ayuda**, haga clic en **Ayuda dinámica**.

Menú Ayuda

Podemos utilizar el menú **Ayuda** para navegar por los documentos de la Ayuda y acceder a numerosas categorías de la misma, incluyendo las siguientes:

- **Contenido**

Muestra la ventana Contenido, que permite navegar por la tabla de contenido de MSDN Library.

- **Índice**

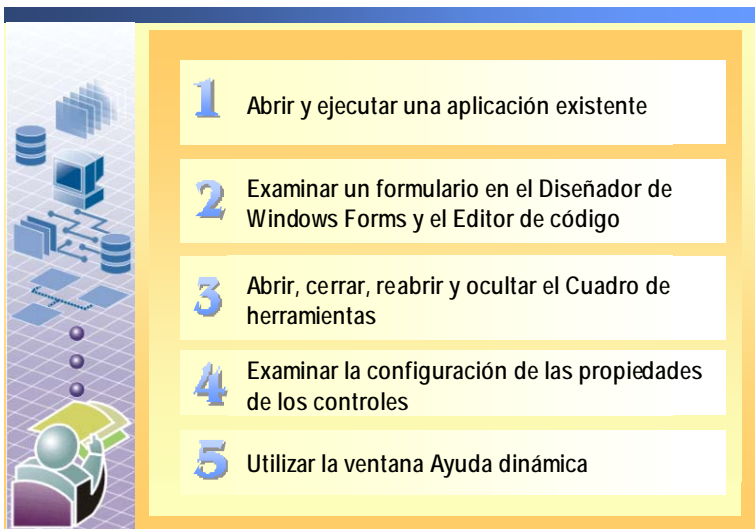
Muestra la ventana Índice, que permite escribir palabras clave para localizar temas en MSDN Library.

- **Búsqueda**

Muestra la ventana Búsqueda, que permite escribir texto y buscarlo en MSDN Library.

También podemos acceder a estas categorías haciendo clic en los botones de la barra de herramientas de la ventana Ayuda dinámica.

Práctica: Trabajar con el entorno de desarrollo



Introducción

En esta práctica, aprenderemos a utilizar las herramientas de programación del entorno de desarrollo de Visual Studio .NET.

Abrir un proyecto

🔍 Abrir y ejecutar una aplicación existente en Visual Basic .NET

1. Abra Visual Studio .NET.
Si Visual Studio.NET ya está abierto, en el menú **Archivo**, haga clic en **Cerrar solución** para cerrar todas las soluciones abiertas.
2. En el menú **Archivo**, seleccione **Abrir** y, a continuación, haga clic en **Proyecto**.
3. Abra la solución ApplicationToExplore.sln de la práctica anterior o del fichero practs02.zip.
4. En la barra de herramientas estándar, haga clic en el botón **Iniciar** ▶ o pulse la tecla F5.
5. Haga clic en los botones del formulario para ver cómo funciona la aplicación.
6. Salga de la aplicación que se está ejecutando.

Examinar vistas de formulario

✎ Examinar un formulario en el Diseñador de Windows Forms y el Editor de código

1. En Explorador de soluciones, haga clic con el botón derecho en **Form1.vb** y, a continuación, haga clic en **Ver diseñador**. Responda a las siguientes cuestiones:

- a. ¿Qué ocurre?

Se abre Form1.vb[Diseño].

- b. ¿Sigue estando disponible la Página de inicio?

Sí, ya que la pestaña de la Página de inicio todavía está visible.

2. Haga clic con el botón derecho en cualquier parte del formulario (excepto en la barra de título) y, a continuación, haga clic en **Ver código** para cambiar al Editor de código (observe que **Ver código** también está disponible desde el menú de acceso directo de *Form1.vb* en el Explorador de soluciones).
3. En el siguiente espacio, escriba los nombres de los gestores de eventos que se muestran en el Editor de código.



Private Sub _____ & Private Sub _____

Button1_Click y Button2_Click

4. Haga clic en la pestaña **Form1.vb[Diseño]** para volver a la vista de diseño.

Uso del Cuadro de herramientas


✎ Abrir, cerrar, reabrir y ocultar el Cuadro de herramientas

1. Seleccione el icono del Cuadro de herramientas  del borde izquierdo de la ventana del entorno. Se abrirá el Cuadro de herramientas.
2. Haga clic en el icono del pin  de la barra de título del Cuadro de herramientas. ¿Qué ocurre?

El Cuadro de herramientas está acoplado, y sigue abierto.

3. Cierre el Cuadro de herramientas.

4. Para volver a abrir el Cuadro de herramientas, en el menú **Ver**, haga clic en **Cuadro de herramientas**.

Observe que el icono con el pin ha cambiado. Haga clic en el icono con el pin  de la barra de título del Cuadro de herramientas de nuevo. ¿Qué ocurre?

La configuración de ocultación automática del Cuadro de herramientas se pone en activo.

Consejo Si hace clic en un control accidentalmente, simplemente pulse ESC dos veces.

Examinar propiedades

✚ Examinar la configuración de las propiedades de los controles

1. En la vista de diseño, haga clic en **Button1**. Examine la configuración de las propiedades **Name** y **TabIndex**.
2. Haga clic en cualquier parte del formulario salvo sobre un control. ¿Qué ocurre en la ventana Propiedades?

Las propiedades del formulario aparecen en la ventana Propiedades.


3. Examine las propiedades **Name** y **BackgroundImage** del formulario.

Uso de la Ayuda

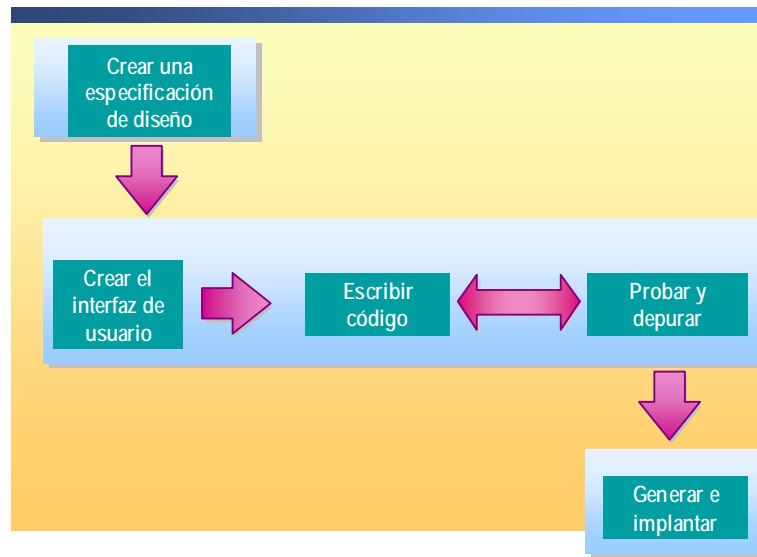
✚ Utilice la ventana Ayuda dinámica

1. Verifique que Form1.vb se muestra en la ventana de diseño. Haga clic en el formulario, haga clic en **Button1** y, a continuación, haga clic en la propiedad **BackColor** para Button1.
2. Haga clic en la pestaña **Ayuda dinámica** de la parte inferior de la ventana Propiedades. Si la pestaña no está visible, en el menú **Ayuda**, haga clic en **Ayuda dinámica**.

Observe que los artículos propuestos están relacionados con los elementos seleccionados en la ventana Propiedades.

3. En la lista de artículos debajo del encabezado **Ayuda**, haga clic en **Ventana de propiedades** y visualice los temas de ayuda relacionados.
4. En el menú **Ayuda**, haga clic en **Contenido**.
5. En la ventana Contenido, haga clic en **Visual Basic y relacionado** en el cuadro **Filtrado por**.
6. Haga clic en el botón **Sincronizar contenidos**  de la barra de herramientas para ver la ubicación de este tema en el contenido de la Ayuda.
7. Salga de Visual Studio .NET.

Lección: Crear un proyecto Visual Basic .NET



Introducción

Esta lección explica cómo crear una aplicación sencilla en Visual Basic .NET. El objetivo de la lección es introducir al proceso de programación global, no proporcionar información detallada sobre la sintaxis del lenguaje ni la implementación.

Estructura de la lección

Esta lección incluye los siguientes temas y actividades:

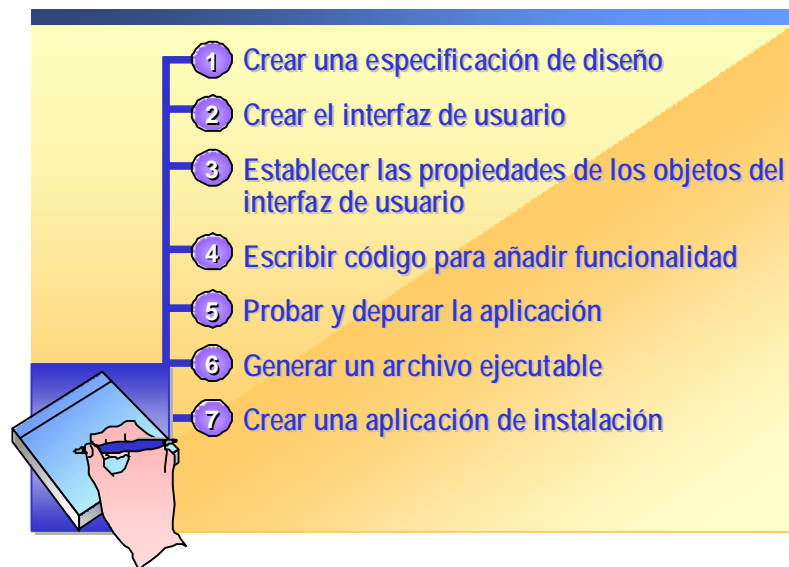
- El proceso de desarrollo
- Cómo crear el interfaz de usuario
- Cómo establecer las propiedades de los controles
- Cómo añadir código a los controles
- Cómo guardar, ejecutar y generar la aplicación

Objetivos de la lección

En esta lección, aprenderá a:

- Describir el proceso de crear una aplicación en Visual Basic .NET.
- Iniciar un nuevo proyecto y crear los objetos del interfaz de usuario.
- Establecer propiedades para los controles.
- Escribir código de eventos para añadir funcionalidades a una aplicación.
- Guardar, ejecutar y crear una versión *release* de una aplicación.

El proceso de desarrollo



Visual Basic .NET contiene todo lo necesario para crear nuestras propias aplicaciones para Windows, de principio a fin. Para crear el interfaz de usuario, ubicamos controles en un formulario desde el Cuadro de herramientas. A continuación, personalizamos los controles estableciendo propiedades. Seguidamente, definimos qué debería hacer nuestro programa escribiendo el código. Finalmente, podemos guardar, ejecutar y compilar nuestro programa para que otros también puedan utilizarlo.

Pasos básicos

Crear una aplicación en Visual Basic .NET implica siete pasos básicos:

1. Crear una especificación de diseño.

La especificación de diseño es el anteproyecto que utilizamos cuando creamos una aplicación. Es bueno tomarnos tiempo antes de escribir código para diseñar la aplicación que vamos a crear. Aunque Visual Basic .NET proporciona herramientas de ayuda para desarrollar una solución con rapidez, tener una idea clara de las necesidades del usuario y el conjunto inicial de características nos ayudará a ser más eficaces en la labor de desarrollo. También nos ayudará a ahorrar tiempo al minimizar la posible reescritura de código debido a una pobre o inexistente especificación de diseño.

2. Crear el interfaz de usuario.

Para crear el interfaz de nuestra aplicación, en primer lugar debemos ubicar controles y objetos en un formulario dibujándolos o pintándolos en el Diseñador de Windows Forms. Podemos observar otras aplicaciones, como Microsoft Excel o Microsoft Word, para obtener ideas sobre cómo diseñar el interfaz. Si necesitamos información sobre el diseño de interfaces, podemos consultar el libro *Microsoft Windows User Experience*, publicado por Microsoft Press®.

3. Establecer las propiedades de los objetos del interfaz de usuario.

Después de añadir objetos a un formulario, podemos establecer sus propiedades en la ventana Propiedades o en el Editor de código.

4. Escribir código para añadir funcionalidad.

Una vez hayamos establecido las propiedades iniciales del formulario y sus objetos, podemos añadir código que se ejecute en respuesta a eventos. Los eventos ocurren cuando se realizan diferentes acciones sobre un control u objeto. Por ejemplo, el evento **Clic** de un botón tiene lugar cuando un usuario hace clic sobre él con el ratón. Para la mayoría de las aplicaciones, también necesitaremos escribir código para añadir lógica de negocio y para acceder a datos.

5. Probar y depurar la aplicación.

Probar y depurar no es un paso que se haga una vez, sino algo que realizamos reiteradamente durante el proceso de desarrollo. Cada vez que realizamos un cambio importante en alguno de los pasos 2, 3 y 4, se ejecutará una versión de depuración de la aplicación y nos aseguraremos de que funciona según lo esperado. Visual Basic .NET ofrece numerosas herramientas de depuración que podemos utilizar para encontrar y reparar errores de su aplicación.

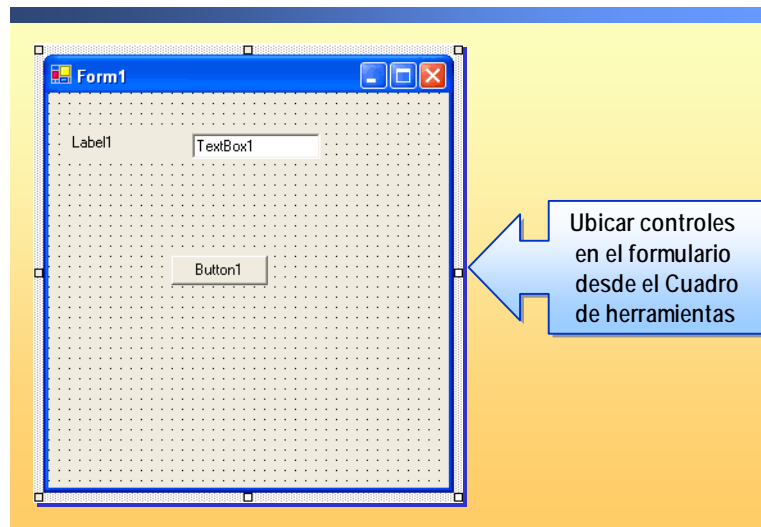
6. Generar un archivo ejecutable.

Una vez finalizado el proyecto, crearemos una versión *release* del proyecto y lo volveremos a probar y depurar. Esto compila los diversos archivos que constituyen el programa en un archivo ejecutable aislado denominado *ensamblado*.

7. Crear una aplicación de instalación.

Para ejecutar nuestra aplicación, normalmente el usuario necesita otros archivos, como cualquier archivo DLL (*dynamic-link library*) que hayamos utilizado para crear nuestra aplicación. Visual Basic .NET proporciona el asistente de instalación (*Setup Wizard*), que automatiza la creación del programa de instalación y garantiza que el usuario tiene todos los archivos necesarios.

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

Introducción

Podemos modificar las propiedades seleccionando objetos en el formulario y cambiando su configuración en la ventana Propiedades.

Establecer las propiedades de control

⚡ Establecer propiedades en la ventana Propiedades

1. Si la ventana Propiedades no está abierta, en el menú **Ver**, haga clic en **Ventana de propiedades**.
2. Haga clic en el control para el que desea establecer una propiedad.
3. En la ventana Propiedades, seleccione la propiedad y establezca el valor deseado.

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
```

Introducción

Una vez establecidas las propiedades iniciales de los objetos del interfaz de usuario, podemos añadir código que ejecute en respuesta a eventos. Los eventos ocurren cuando se realizan diferentes acciones sobre un control u objeto. Por ejemplo, el evento **Clic** de un botón ocurre cuando el usuario hace clic en él con el ratón.

Escribir código

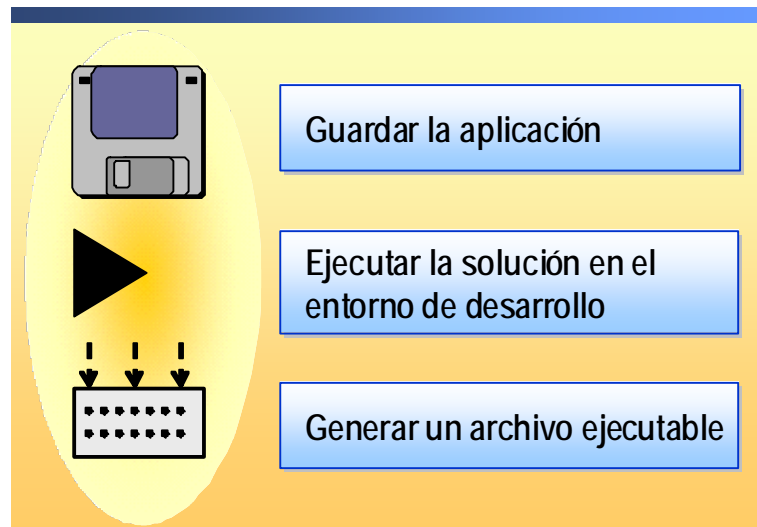
✍ Escribir código para un evento de un control

1. En Explorador de soluciones, seleccione el formulario para el que desea escribir código.
2. En el menú **Ver**, haga clic en **Código** para abrir el Editor de código.
3. En la lista **Nombre de clase**, haga clic en el control para el que desea escribir código.
4. En la lista **Nombre de método**, haga clic en el evento deseado para abrir su gestor de eventos en el Editor de código.

El Editor de código muestra las sentencias de programación que marcan el inicio y el final del procedimiento de este evento en particular.

5. Escriba su código entre los puntos inicial y final del cuerpo del procedimiento.

Cómo 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**.
Cuando 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.