STA. MARÍA LA nueva SAN JOSÉ ARTESANO

Windows Forms en Visual Studio con C#

Creación de una aplicación de Windows Forms en Visual Studio con C#

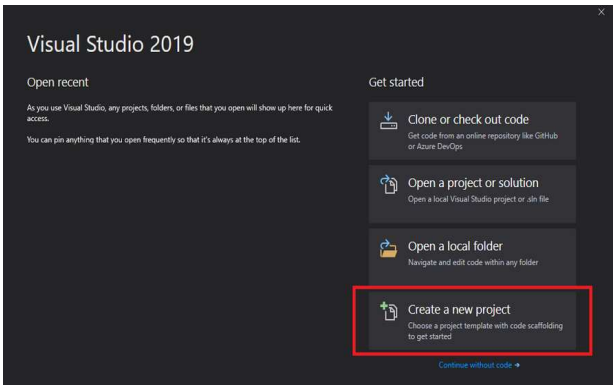
HELLO WORLD!

STA. MARÍA LA nueva SAN JOSÉ ARTESANO

Crear un proyecto

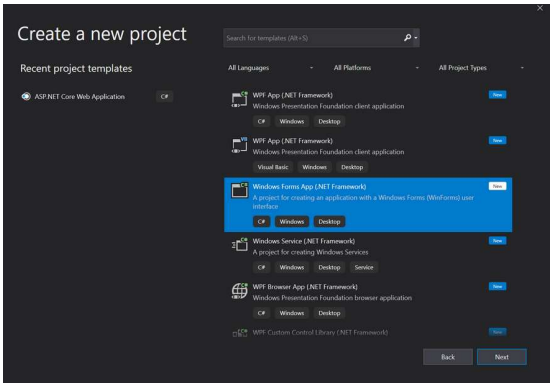
- En primer lugar, se creará un proyecto de aplicación C#. En el tipo de proyecto se incluyen todos los archivos de plantilla que vamos a necesitar, sin necesidad de agregar nada más.

1. Abra Visual Studio.
2. En la ventana de inicio, elija **Crear un proyecto nuevo**.



Crear un proyecto

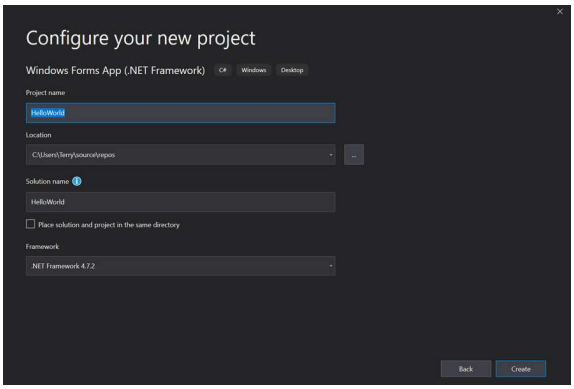
3. En la ventana **Crear un nuevo proyecto**, elija la plantilla **Windows Forms App (.NET Framework)** para C#.



Crear un proyecto

4. En la ventana **Configurar el nuevo proyecto**, escriba *HelloWorld* en el cuadro **Nombre del proyecto**. Luego, elija **Crear**.

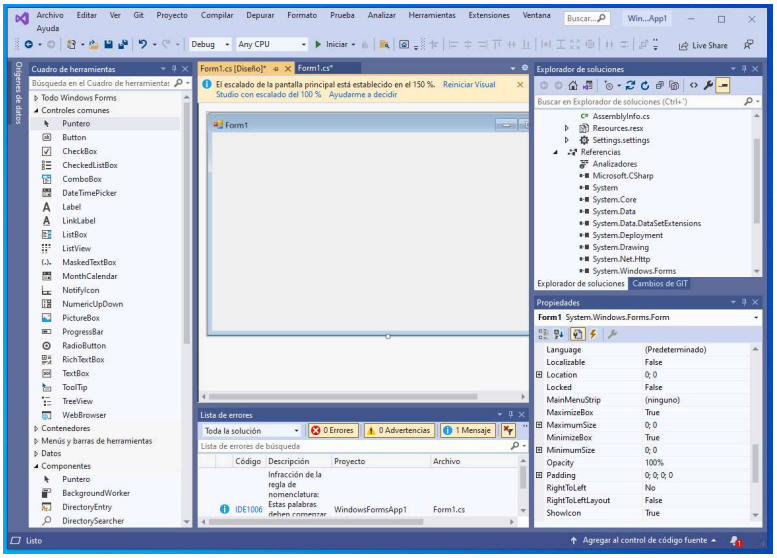
Una observación, si queréis cambiar una carpeta predeterminada donde se almacenen los proyectos:
herramientas→opciones→proyectos y soluciones→ubicaciones y elige la ubicación deseada



Crear la aplicación

- Tras seleccionar la plantilla de proyecto de C# y asignar un nombre al archivo, Visual Studio abre un formulario automáticamente Form1. Un formulario es una interfaz de usuario de Windows.
- Se va a crear una aplicación "Hello World"; para ello, se agregarán controles al formulario y, después, se ejecutará la aplicación.

Entorno de desarrollo



- Cuadro de herramientas.
- Cuadro de diseño.
- Lista de errores.
- Propiedades.
- Explorador de soluciones.

Explorador de soluciones

- Solución: engloba a uno o más proyectos.
- Proyecto: administra los archivos que componen la aplicación.
- Formularios y módulos:
  - Form1.cs →usado por el programador para escribir código
  - Form1.Designer.cs →usado por el diseñador de formularios
  - Form1.resx→hace referencia a los recursos de la aplicación como imágenes…
- Referencias: agrupa las referencias a las bibliotecas de clases usadas.

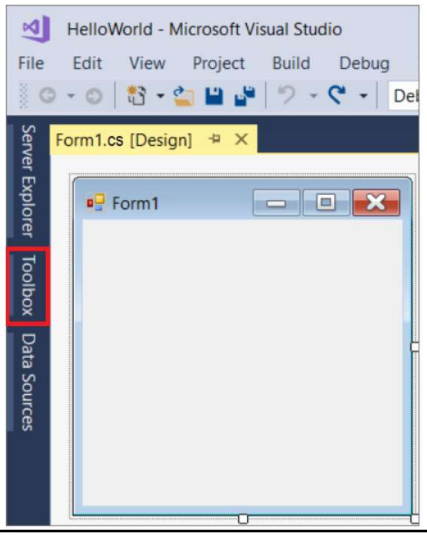
Agregar controles

- En Visual C# tenemos dos tipos de objetos: ventanas o formulario y controles.
- Controles más comunes.
  - **Label** - etiqueta
  - **LinkLabel** – etiqueta para hipervínculos
  - **Button** – botón de pulsación
  - **TextBox** – Caja de texto
  - **MaskedTextBox** – Caja de texto mejorada con control de validación de entrada de texto.
  - **MenuStript** – Barra de menús
  - **CheckBox –** Casilla de verificación
  - **RadioButton –** Botón de opción
  - **GroupBox –** marco para formar grupos de botones
  - **PictureBox –** Caja de imagen
  - **Panel –** Contenedor de controles
  - **FlowLayoutPanel –** Panel que coloca el contenido de forma dinámica
  - **TableLayoutPanel –** Panel dinámico de rejilla
  - **DataGridView –** Tabla para visualizar datos
  - **ListBox –** Lista desplegable
  - **CheckedListBox-** Lista desplegable con casilla de verificación
  - **ComboBox –** Caja de texto + lista desplegable
  - **ListView –** Colección de elementos que se pueden visualizar mediante vistas distintas
  - **TreeView –** Colección jerárquica
  - **TabControl - n**ontenedor
  - **DateTimePicker –** Selecciona fecha y hora
  - **MonthCalendar –** Calendario mensual
  - **HScrollBar y VScrollBar –** Barras de desplazamiento
  - **Timer –** Temporizador
  - **ProgressBar** – Barra de progreso
  - **RichTextBox** – Caja de texto enriquecido
  - **TootlTip** – Descripciones breves
  - **OpenFileDialog, FontDialog, PrintDialog** – Cajas de diálogo

Agregar un botón al formulario

1. Elije **Cuadro de herramientas** para abrir la ventana flotante Cuadro de herramientas.

Ver > **Cuadro de herramientas**. También puede presionar **CTRL+Alt+X**.

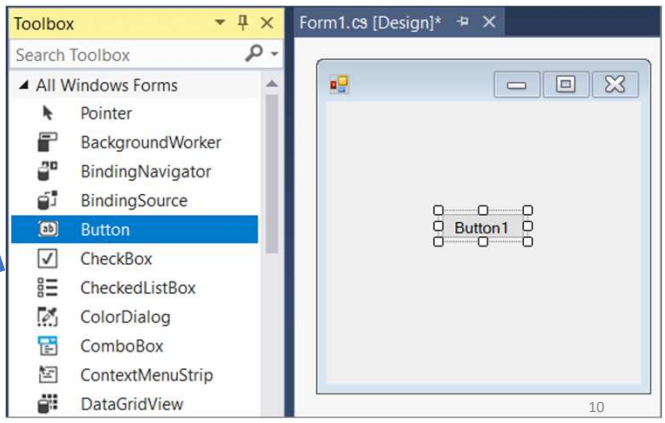


Agregar un botón al formulario

2. Elije el icono **Anclar** para acoplar la ventana **Cuadro de herramientas**.

3. Elija el control **Botón** y arrástrelo al formulario.

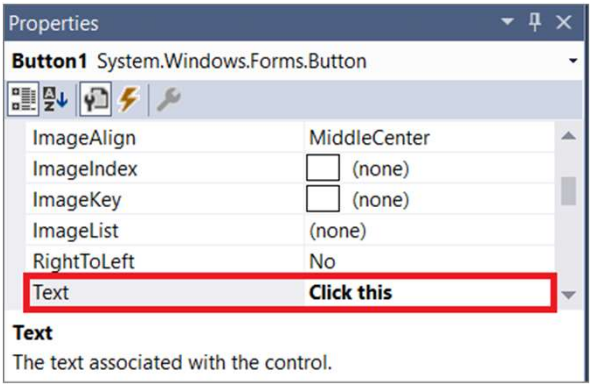
Al colocar el control aparecen una líneas indicando la alineación. Puedes elegir entre los modos:
SnapLines – Líneas de ayuda
SnapToGrid – Rejilla
Desde Herramientas→Opciones→Diseñador Windows Forms→Modo de diseño



Propiedades de un control de botón

4. En la ventana Propiedades, busca Texto, cambia el nombre de Button1 a Click this y, luego, presione Entrar.

Si no ves la ventana **Propiedades**, puede abrirla desde la barra de menús. Para ello, elija **Ver** > **Ventana Propiedades**. También puede presionar **F4**.



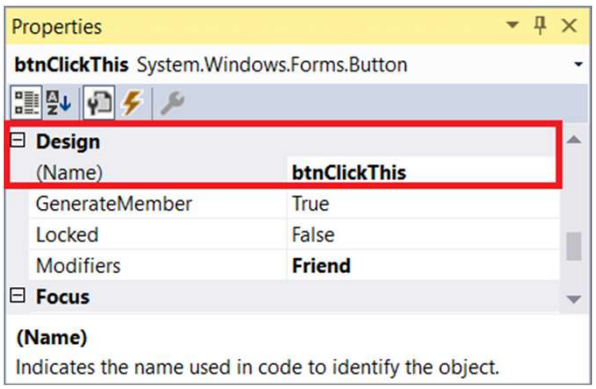
Propiedades de un control de un botón

5. En la ventana Propiedades, cambia el nombre de Button1 a btnClickThis y, tras ello presiona Entrar.

Cada control tiene una serie de propiedades (atributos).

Cada propiedad de un objeto tiene un valor por defecto que puede ser modificado.

También se puede modificar el valor de una propiedad durante la ejecución de la aplicación: lblHelloWorld.Text = "Hello World!";



Agregar una etiqueta al formulario

Ya hemos agregado un control de botón para crear una acción, así que ahora vamos a agregar un control de etiqueta al que enviar texto.

1. Selecciona el control Etiqueta desde la ventana Cuadro de herramientas, arrástralo hasta el formulario y colócalo debajo del botón Click this.
2. En la ventana Propiedades, cambie el nombre del Lavel1 a lblHelloWorld y presione Entrar.

Una vez puestos todos los controles y ajustado su tamaño puedes bloquearlos para no moverlos accidentalmente desde: Formato → Bloquear controles

Escribir controladores de eventos

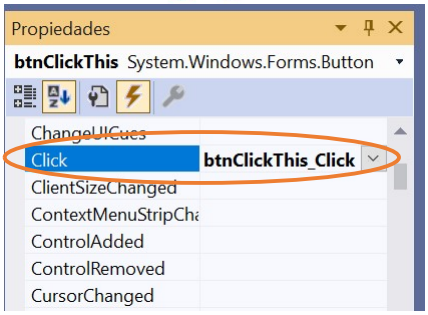
Cuando sobre un objeto ocurre un suceso se produce un evento.

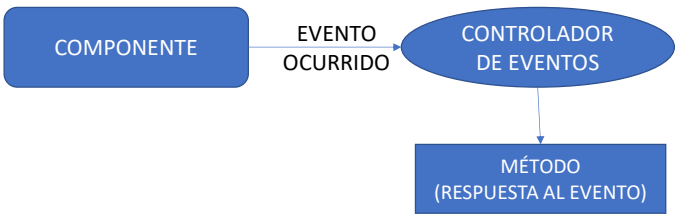
Ej: Un usuario hace clic **(evento)** sobre un botón **(objeto)**.

Si queremos que nuestra aplicación responda al evento tenemos que escribir un método **(controlador de eventos)**.

Este método pertenece al objeto padre (Formulario).

La lista de eventos se ve en la ventana propiedades.





Escribir controladores de eventos

1. En la ventana Form1.cs [Diseño] , haz doble clic en el botón Click this para abrir la ventana Form1.cs.
(También puedes expandir Form1.cs en el Explorador de soluciones y luego elegir Form1).

2. En la ventana Form1.cs, después de la línea private void, escribe **lblHelloWorld.Text = "Hello World!";**

Escribir los controladores de eventos
En C# (como en cualquier lenguaje OO) La forma de acceder a una propiedad es:
Objeto.Propiedad

Form1.cs | Form1.cs [Design]

HelloWorld | HelloWorld.Form1

```
1   using System;
2   using System.Windows.Forms;
3
4   namespace HelloWorld
5   {
6       public partial class Form1 : Form
7       {
8           public Form1()
9           {
10              InitializeComponent();
11          }
12
13          private void btnClickThis_Click(object sender, EventArgs e)
14          {
15              lblHelloWorld.Text = "Hello World!";
16          }
17      }
18  }
19
```

El primer parámetro del método, sender, hace referencia al objeto que generó el evento.
El segundo parámetro, e, contiene información que depende del evento.

Escribir controladores de eventos

- En Form1.Designer.cs se genera el siguiente código

El método btnClickThis_Click controla (EventHandler) el evento Click de btnClickThis

Form1.Designer.cs | Form1.cs | Form1.cs [Diseño]

HelloWorld | HelloWorld.Form1 | InitializeComponent()

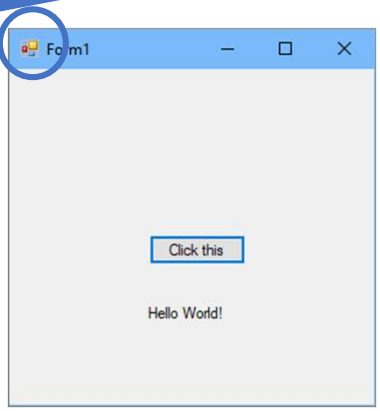
```
35              //
36              // btnClickThis
37              //
38              this.btnClickThis.Location = new System.Drawing.Point(363, 129);
39              this.btnClickThis.Name = "btnClickThis";
40              this.btnClickThis.Size = new System.Drawing.Size(94, 40);
41              this.btnClickThis.TabIndex = 0;
42              this.btnClickThis.Text = "Click this";
43              this.btnClickThis.UseVisualStyleBackColor = true;
44              this.btnClickThis.Click += new System.EventHandler(this.btnClickThis_Click);
45              //
```

16

Ejecutar la aplicación

Puedes incluir tu propio icono (archivo .ico) de 16x16 o 32x32 pixeles en la propiedad Icon de Formulario

1. Para ejecutar **Depurar→Iniciar sin depurar o Ctrl+F5.**
2. Haz clic en **Click this** en el cuadro de diálogo **Form1**. El texto **lblHelloWorld** cambia a **Hello World!** .



PROGRAMA CONDUCIONDO POR EVENTOS Y ORIENTADO A OBJETOS

RESÚMEN: Una aplicación de Interfaz de Windows Forms usa objetos (ventanas) donde se insertan otros objetos (controles) y finalmente se escribe el código relacionado con la funcionalidad de ese objeto.