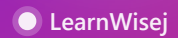


Primeros pasos con Wisej.NET

De un Visual Studio vacío a una aplicación en ejecución, y los hábitos que mantienen el código limpio.



Objetivo de la lección

Crea un primer proyecto Wisej.NET, ábrelo en el Designer, ejecútalo en local, examina la estructura de la solución y aprende dónde están los archivos de inicio y de configuración.

1. Empieza con la plantilla de proyecto

En Visual Studio, crea un nuevo proyecto a partir de las plantillas de Wisej.NET instaladas. La lista exacta varía según la versión, pero el flujo es el mismo: **crear un nuevo proyecto** → **buscar «Wisej.NET»** → **elegir una plantilla de aplicación web Wisej.NET** → **darle un nombre** → **crear**.

Nombre sugerido para el proyecto de práctica

WisejTrainingApp



Antes de empezar

Asegúrate de que la extensión de Wisej.NET para Visual Studio y sus plantillas están instaladas. Si las plantillas no aparecen, revisa la instalación o reinicia Visual Studio.

2. Abre el Designer de Wisej.NET

El Designer es donde colocas los controles de forma visual. Abre la página o ventana principal y cambia a la vista de diseño; después arrastra controles desde el Toolbox, establece propiedades y haz doble clic en un control para crear un manejador de eventos.

Tarea en el Designer	Qué haces
Añadir un control	Arrastra un <code>Button</code> , <code>Label</code> , <code>TextBox</code> , <code>Panel</code> o <code>DataGridView</code> a la superficie.
Establecer el texto	Selecciona el control, abre Properties y cambia la propiedad <code>Text</code> .
Renombrar un control	Cambia <code>Name</code> por algo claro, como <code>btnRun</code> , <code>lblStatus</code> , <code>txtCustomerName</code> .
Crear el código del clic	Haz doble clic en un botón (o usa el panel Events) para crear un manejador <code>Click</code> .
Ajustar el diseño	Usa <code>Dock</code> , <code>Anchor</code> , <code>Size</code> , <code>Location</code> , <code>Margin</code> , <code>Padding</code> .

3. Ejecútalo en local

Ejecutar en local significa probar en tu propio equipo antes de publicar. Usa el botón Start / Debug habitual: el proyecto se compila, se inicia un servidor local y la aplicación se abre en el navegador.

1. Haz clic en **Start / Debug** en Visual Studio.
2. Espera a que el proyecto se compile.
3. Confirma que el navegador abre la aplicación Wisej.NET.
4. Prueba el clic de un botón para confirmar que tu código se ejecuta.
5. Usa puntos de interrupción para inspeccionar los manejadores de eventos y las variables.



Hábito de diagnóstico

Si la aplicación no se abre, lee primero las ventanas **Error List** y **Output** de Visual Studio. La mayoría de los problemas de principiante son paquetes que faltan, errores de compilación, un proyecto de inicio equivocado o un archivo renombrado cuyas referencias no se actualizaron.

4. Examina la estructura de la solución

El Solution Explorer no es un montón de archivos al azar: muestra cómo está organizada la aplicación.

Archivo o carpeta	Para qué sirve
<code>Program.cs</code>	Código de inicio que arranca la aplicación y a menudo muestra la primera Page o Window.
<code>Page1.cs</code> / <code>Window1.cs</code>	Code-behind: donde escribes los manejadores de eventos y el comportamiento de la pantalla.
<code>*.Designer.cs</code>	Diseño e inicialización de controles generados por el Designer. No lo edites a mano.
<code>*.resx</code>	Archivos de recursos para el Designer y los datos localizados.
Archivos de configuración	<code>Web.config</code> y la configuración JSON específica de Wisej.NET.
<code>Models/</code>	Clases de datos sencillas, como <code>Ticket.cs</code> o <code>Customer.cs</code> .
<code>Services/</code>	Lógica de negocio, como <code>TicketService.cs</code> .

5. Buenos hábitos de desarrollo

Hábito	Evita	Prefiere
Nombra los controles con claridad	<code>button1</code> , <code>label2</code>	<code>btnSaveTicket</code> , <code>lblStatus</code>
Mantén legibles los eventos	200 líneas en <code>btnSave_Click</code>	Llamar a <code>ValidateInput()</code> y a <code>ticketService.Save(ticket)</code>
Separa la lógica de negocio	Toda la lógica de guardado en la página	Un <code>TicketService</code> se encarga de las reglas
Organiza los archivos	Todo en la raíz del proyecto	Carpetas Models, Services, Pages, Controls

6. Ejemplo: un manejador de eventos legible

Un manejador de eventos debe leerse como una historia breve: obtener la entrada, llamar a la lógica y actualizar la interfaz:

```
private void btnSayHello_Click(object sender, EventArgs e)
{
    string name = txtName.Text.Trim();

    if (string.IsNullOrEmpty(name))
    {
        lblStatus.Text = "Please enter a name.";
        return;
    }

    lblStatus.Text = $"Hello, {name}!";
}
```

A medida que la aplicación crece, traslada las reglas y las operaciones de datos a clases de servicio:

```
private readonly TicketService ticketService = new TicketService();

private void btnSaveTicket_Click(object sender, EventArgs e)
{
    Ticket ticket = ReadTicketFromScreen();
    ticketService.Save(ticket);
    lblStatus.Text = "Ticket saved.";
}
```



Ya estás listo para el laboratorio

A continuación: mira el recorrido guiado de construcción, supera la evaluación de conocimientos y después crea y ejecuta tu propia primera pantalla Wisej.NET en el laboratorio práctico.