

El formulario

Hasta ahora, todos los libros y manuales que os encontréis donde se habla de programación, explican la programación haciendo uso de la consola en sus ejemplos. La razón es la simpleza de la consola, ya que es la forma más básica de mostrar la salida de un programa.

No obstante, todos somos conscientes de que todas, o la mayoría de las aplicaciones que usamos, tienen una **interfaz gráfica** en lugar de una ventana negra que a mucha gente le asusta porque no la entiende.

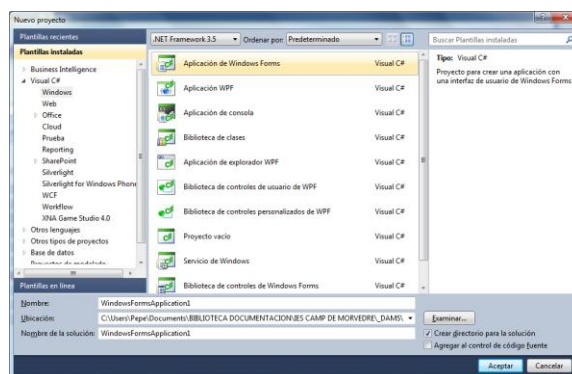
La cuestión, es que para poder crear una interfaz gráfica, hay primero que entender como funciona el lenguaje, ya que normalmente crear la interfaz requiere hacer llamadas a librerías externas, invocar clases, y realizar una serie de tareas bastante complejas...

En el caso de .NET, si usamos Visual Studio o Sharp Develop, o algún otro entorno que nos lo permita, estamos de enhorabuena. Pues lo que es **el formulario se genera automáticamente, y encima podemos crear los elementos del formulario de forma gráfica, simplemente arrastrando los elementos a un formulario. De modo que hasta un niño pequeño podría desarrollar así un programa.**

No obstante, luego hay que “codear” todo el programa. Y eso es lo que hay que llevar entendido de antes. No se puede empezar la casa por el tejado.

Cuando programamos una GUI (lo que es la interfaz) en Windows, ésta se compone de la **ventana**, los **controles** (botones, cajas de texto, cajas de imagen, etc), y una serie de controles que no vemos pero que están ahí, que llamaremos **controladores**.

Sobre los tipos de aplicación que podemos generar para Windows, hay muchos. Muchos que, incluso ahora, no podemos ver porque no tenemos instalado el software necesario. Pero generalmente, usamos 3. Una es la aplicación Windows normal (**Aplicación de formulario de Windows**), que es la que todos conocemos más o menos. Otra es la **librería de clases**, que es una extensión de programa que podemos acoplar a cualquiera de nuestros programas en .net (una librería .dll). Y por último, el **Control de formulario**, que es una dll que podemos acoplar a un formulario .net.

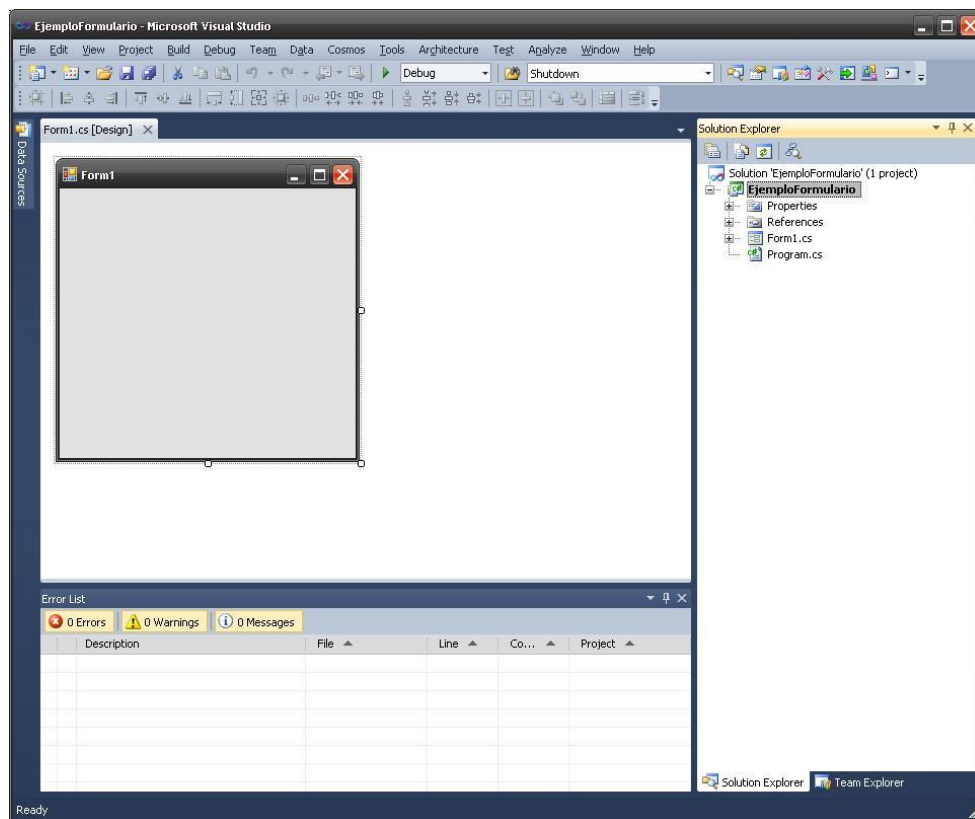


Nuevo Proyecto 1

Adicionalmente tenemos los **servicios de windows**, que son como aplicaciones que se ejecutan en segundo plano, sin que nosotros notemos nada, pero que sirven de apoyo de algún modo a nuestras aplicaciones.

Aunque no lo veamos porque no tenemos el software requerido instalado, podemos también crear aplicaciones con Open GL (requiere una librería específica) o Direct X / XNA para crear aplicaciones en 2D-3D, o incluso sistemas operativos usando el paquete Cosmos.

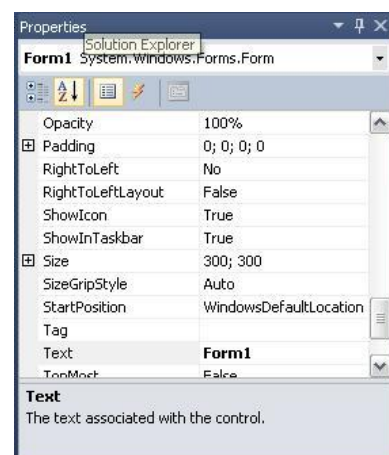
Si creamos un nuevo proyecto de formulario (empezaremos por ahí), veremos que en la pantalla aparece por defecto el **diseñador de formularios**.



Editor de Formularios 1

Desde aquí es desde donde se comienza a diseñar el formulario. No obstante, primero hay que configurar un poco el entorno, pues aún no podemos ver las herramientas necesarias.

Hacemos click en Ver > Caja de herramientas (view > Ttoolbox), y Ver > Propiedades (View > Properties). Aparecerán dos paneles. Uno a la izquierda, que es una lista con todos los controles disponibles, y la otra debajo del explorador de soluciones, que son las propiedades de cada objeto.



Propiedades 1

Si seleccionamos el formulario, haciendo click sobre él, podemos ver todas las propiedades del mismo (y modificarlas).(Esquina inferior derecha)

Podemos cambiar el tamaño, la posición, la opacidad, el icono, el título (Text), etc.

En el panel de la caja de herramientas (Toolbox), disponemos de un amplio set de controles que podemos añadir a nuestro formulario, organizados por categorías. Si desplegamos la pestaña “Todos los controles de Windows” (All windows forms), podremos ver todos los controles que tenemos.

Los más comunes están en “**Controles comunes**” (common controls), y luego hay una serie de controles bastante interesantes llamados “**contenedores**” (containers), que sirven para almacenar controles. Son bastante útiles a la hora de hacer “bonita” una aplicación. Luego tenemos los “**menús y cajas de herramientas**” (menus and toolbars), y por último lo que yo llamo controladores, en “**Componentes**” (components). También tenemos cajas de diálogo y... en fin, ya los iré explicando poco a poco. De momento, id explorando vosotros sin miedo.

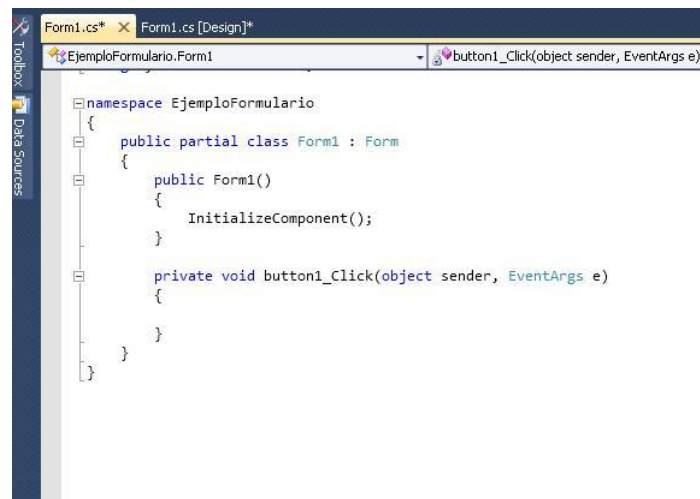
Para añadir un control al formulario, basta con arrastrarlo desde la caja de herramientas al formulario. Por ejemplo, añadimos un botón y una caja de texto:



Diseño 1

Al seleccionar cada elemento, podemos cambiar sus propiedades desde la caja de propiedades. Cambiamos el texto del botón, y si queréis experimentar un poco (recomendado), su estilo y sus colores.

Si hacemos doble click sobre un control, pasaremos a la ventana de código, donde dispondremos de un método para cambiar su código de comportamiento al interactuar con él. Por defecto, si hacemos doble click sobre el botón, nos creará un método o función que se ejecutará al hacer click sobre el botón.



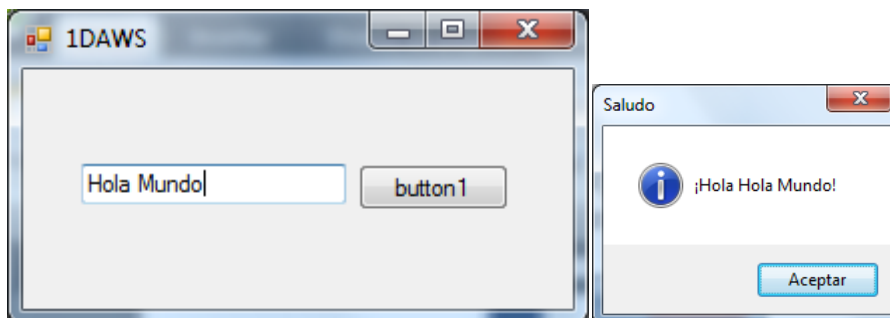
codigo fuente 1

¡Perfecto! No nos tenemos que preocupar de nada, hasta aquí.

En el código del botón (button1_Click), añadimos la siguiente línea:

```
MessageBox.Show("¡Hola " + textBox1.Text + "!", "Saludo", MessageBoxButtons.OK, MessageBoxIcon.Information);
```

Guardamos todo, compilamos y ejecutamos. Probamos a meter un nombre en la caja de texto, y pulsamos el botón. Aparecerá un mensaje:



Resultado de la ejecución 1