

# Conociendo el entorno visual

Sitio: Agencia de Habilidades para el Futuro  
Curso: Desarrollo de Sistemas Orientado a Objetos 1º D  
Libro: Conociendo el entorno visual

Imprimido por: Eduardo Moreno  
Día: domingo, 23 de marzo de 2025, 10:01

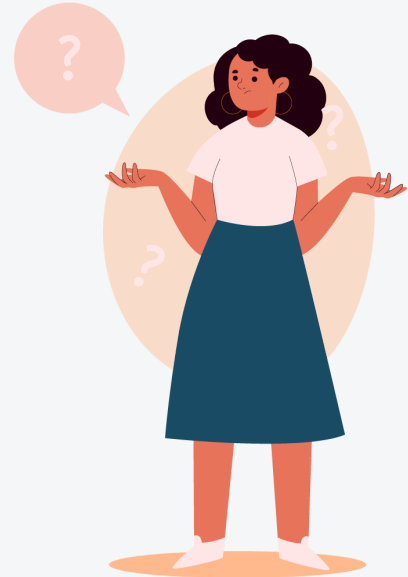
# Tabla de contenidos

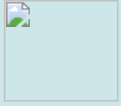
- 1. Preguntas orientadoras**
- 2. Entorno visual**
- 3. IDE visual en C# V2**
- 4. Formulario**
- 5. Controles básicos**
  - 5.1. Creación del diseño
- 6. Cuadro de diálogo**
  - 6.1. Creación del cuadro de diálogo
- 7. Como recuperar datos de un TextBox**



## Preguntas orientadoras

- ¿Arrastrar o copiar controles?
- ¿Los controles que se usan en el diseño, los tengo que dibujar?
- ¿Cuántos controles puedo colocar?
- ¿Un control se vincula con sus propiedades?





## Entorno visual

Comenzaremos a dar algunos primeros pasos para la programación reconociendo nuestro entorno visual. Pero, antes, ¿sabés de qué se trata? Este estilo de programación presenta las siguientes características:

*Hacé clic sobre las imágenes de la derecha para recorrer las características.*

 **Entorno Visual**  
Características





Sigamos recorriendo ...

A continuación, se presenta el entorno de trabajo para el desarrollo de una solución en C#.

*Hacé clic sobre la flecha para deslizarla de una imagen a otra.*





## El Formulario

La codificación en entorno visual contiene dos áreas de trabajo que son: el área de diseño y el área de codificación.

Al trabajar en el área de diseño usamos los controles que nos ofrece la barra de herramientas, generalmente pinchada en el lateral izquierdo del IDE de programación, arrastrándolos hasta el contenedor que es el **formulario**.

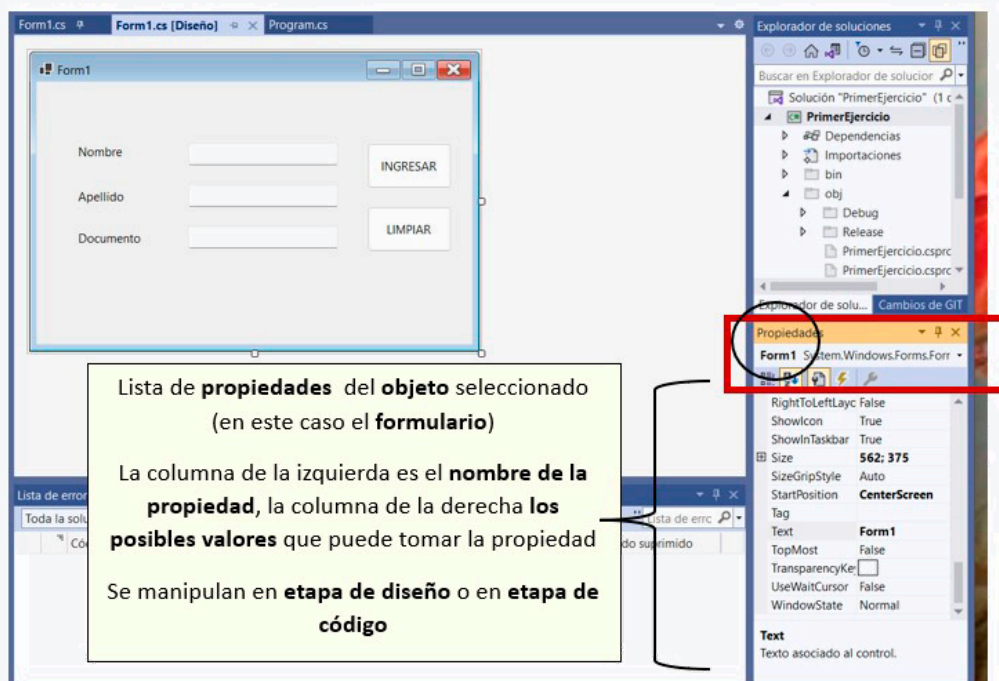
El formulario es una **clase** dentro del **namespace** de la **solución**. La solución es nuestra aplicación, ejercicio (dependiendo en que momento estamos de la codificación) que es reconocido con el nombre que asignamos al momento de iniciar el Visual Studio y creamos el proyecto.

Cuando al formulario que pensamos usar le damos un nombre, lo que estamos haciendo es **instanciar** a un elemento de esa clase y, como viste en la explicación de los contenidos, se genera un **objeto**.

Ese objeto tiene propiedades o características que nos permite personalizarlo. En el IDE de programación lo manejamos desde las ventanas que nos ofrece.

### ¡Visualicemos en pantalla!

Observemos la siguiente imagen:



### ¿Qué es lo que está sucediendo en pantalla?

Recorramos paso a paso:

- 1) Cada uno de los controles que colocamos en el formulario al igual que este tiene propiedades que comparten y propiedades exclusivas. Lo primero que debés hacer es darle un nombre, de esta forma se identifica en el código que función tiene el control.
- 2) La mayoría de lenguajes de programación cuenta con una comunidad de programadores, la conexión con la comunidad nos ayuda a solucionar errores en la ejecución. Tienen algunas reglas o normas para que más allá del lugar de procedencia todos puedan colaborar. Una de las reglas es utilizar prefijos para identificar al control.

3) Estos prefijos se escriben en minúscula y a continuación y sin espacios se escribe el nombre que le queremos dar al control colocando en **mayúscula la primera letra**.

4) Una vez que instanciamos a un elemento de la clase comenzamos a manipular sus propiedades. La propiedad `text` permite escribir leyendas que contenga el control, cuenta también con formatos de letras, color de tipografía, color de fondo, ubicación en la pantalla etc.



La lista de los prefijos está en el documento **PREFIJOS Controles.pdf**.



Te proponemos ser curioso/a e investigar cada propiedad mientras codificas, se puede visualizar en el momento como responde en la ejecución.



## Controles básicos

Tal como nos indica nuestro manual Visual Studio, los controles **son objetos y los objetos, como hemos visto, son combinaciones de datos y códigos que se tratan como una unidad** .



**Relacionemos con un ejemplo observando el siguiente enunciado:**

*"En un instituto terciario deben registrar el nombre, apellido y documento de los futuros postulantes para luego asignarles el curso y procesar la inscripción ..."*

**¿Qué observamos?** El enunciado culmina con puntos suspensivos porque lo iremos completando a medida que avancemos con las explicaciones semanales.

El diseño propuesto puede ser similar a este:

Nombre	.....	INGRESAR
Apellido	.....	LIMPIAR
Documento	.....	

**Nombre, Apellido y Documento** son controles que solamente muestran leyendas, mientras ejecutamos el programa no pueden recibir desde el exterior (o sea desde teclado) el ingreso de caracteres. Este control se llama **label** y su prefijo es **lbl**

Los **puntos suspensivos** simulan el ingreso de datos desde el teclado, el control que permite guardar lo que queremos escribir se llama **textBox** y su prefijo es **txt**

**Ingresar y Limpiar** dan idea de acciones que permiten, la primera, almacenar el nombre, apellido y documento y la segunda, borrar con la escritura que muestre la pantalla.

El botón se llama **button** y su prefijo es **btn** , y la acción o evento predeterminado que se ejecuta para que despliegue las líneas de código es el evento **click**

En resumen: son acciones que se ejecutan independientemente y el control que las muestra es un botón. Este control es muy común de ver en la navegación en la web, siempre cliqueamos en alguno para que suceda algo.

**A continuación, ¡pongamos en práctica!**





## Creación del diseño

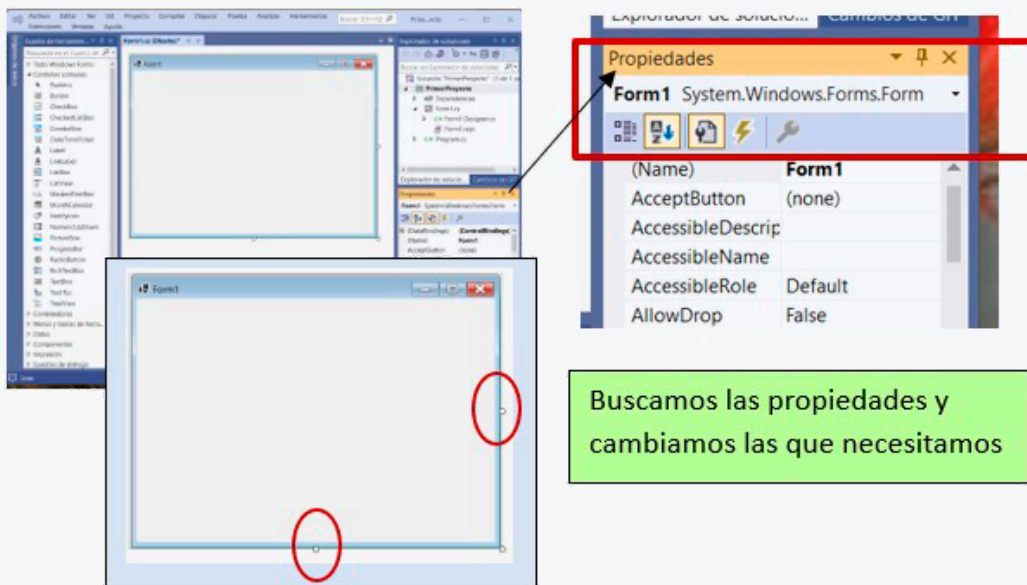


### ¡Manos a la obra!

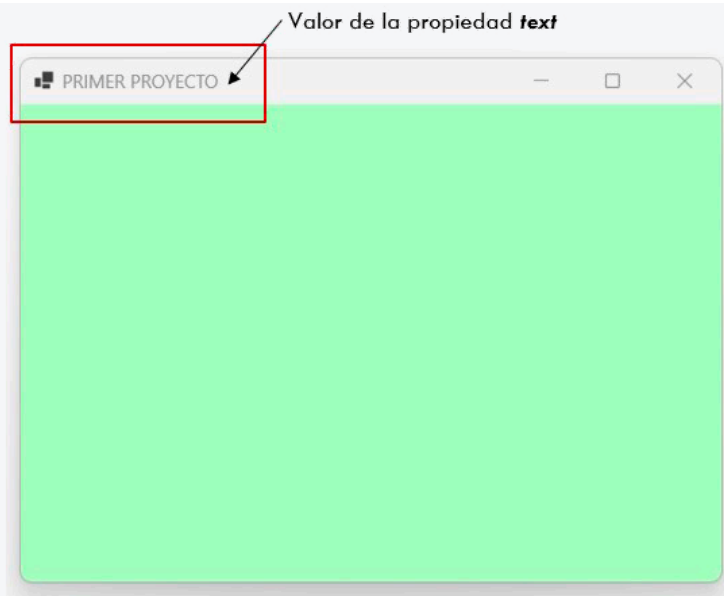
Usaremos un formulario, 3 label, 3 textBox y 2 button; las características se detallan en la actividad 1.

A continuación, te detallamos lo que deberás hacer:

- Abrir Visual Studio 2022 y crear una aplicación de Windows forms llamada **PrimerProyecto**.
- Cuando aparece el IDE, seleccionar el formulario y buscar la ventana **Propiedades**.
- Modificar las propiedades Name, Text, StartPosition y BackColor. La primera instancia la clase, la segunda cambia el nombre en el extremo superior del formulario, la tercera permite posicionar el formulario cada vez que se ejecuta apareciendo en el centro de la pantalla y la última se colorea el fondo del control.

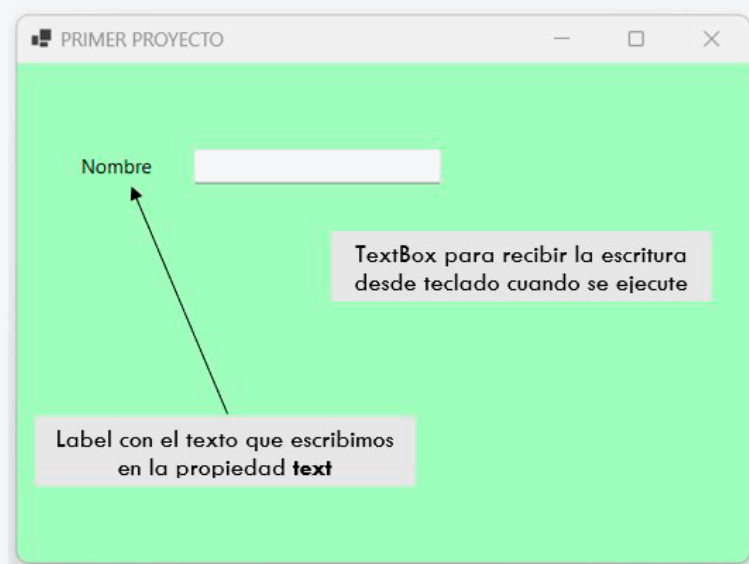


- Para ejecutar y ver como va quedando vamos a la barra de herramienta a **Depurar** y en el listado buscamos **Iniciar Depuración** o hacemos click en el símbolo de play de la barra superior.
- El formulario aparece en el centro de nuestra pantalla con un diseño similar al siguiente:

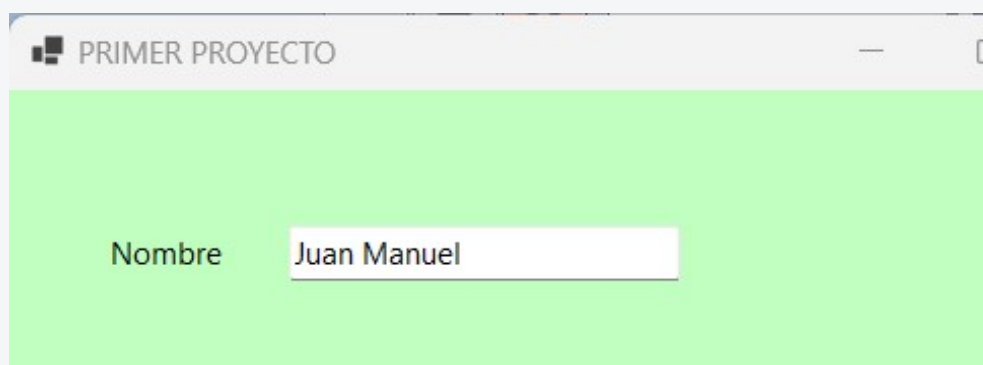


- Colocaremos un label y un textBox. Ambos controles se arrastran al formulario y los colocamos según el diseño propuesto. Las propiedades que vamos a cambiar son Name y Text en ambos.

La imagen debe quedar de la siguiente manera:



Observá cómo al ejecutar permite el ingreso de datos:



**¡Importante!**

El resto del diseño está explicitado en la "Creación de la primera solución en el entorno visual".



## Cuadro de diálogo

Otro gran desafío en el conocimiento del entorno visual, sin duda, refiere a la creación de cuadros de diálogo. Si buscamos que es un cuadro de diálogo seguramente nos encontramos con la siguiente definición

Un cuadro de diálogo es una ventana secundaria que permite a los usuarios/as realizar un comando, formular a estos/as una pregunta o proporcionarles información o comentarios de progreso.

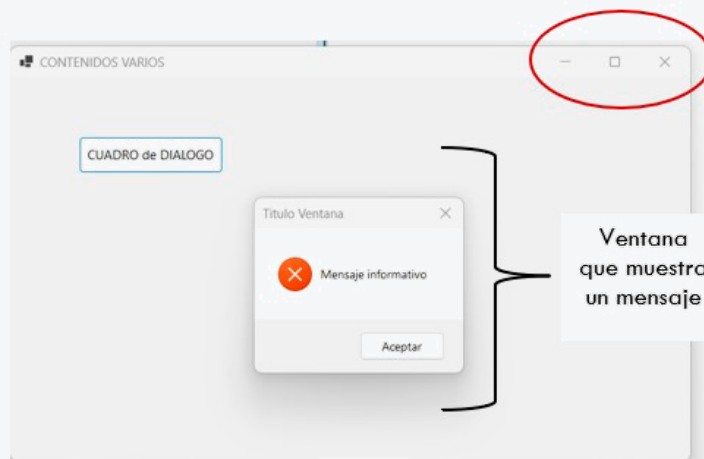
### Amplíemos las características de los cuadros de diálogos

Es importante que profundicemos en la conceptualización de los cuadros de diálogos y definir como interactúan en el entorno visual.



#### Relacionemos con un ejemplo

Observa la siguiente imagen:



El formulario presenta una ventana delante con información. La presencia de esta ventana no permite que el formulario reciba alguna acción externa, lo deja congelado.

Pero, ¿Qué significa? Lo que nos quiere decir la ventana es que **primero** debemos prestarle atención a ella, clicar en alguno de los botones que contiene o en el único como es este caso. Obviamente, debemos leer la información que nos muestra. En conclusión, **es una salida de información desde la aplicación.**



## ¿Cómo se crea un cuadro de diálogo?

Es una función con parámetros, cada uno de ellos representa una parte de la ventana, y, generalmente, aparece cuando otro control recibe el evento click.

El cuadro fue programado en el button con la leyenda **Cuadro de Dialogo**.

Para escribir líneas de código en el entorno visual debemos contar con un procedimiento asociado a un objeto o un procedimiento escrito que luego es invocado.

Si el procedimiento está asociado a un objeto, este a su vez se vincula a un **evento**, es decir, a una acción que se ejerce sobre él. En nuestro ejemplo, las líneas de código se escriben para el objeto button en su evento click. Por el momento, el procedimiento pertenece al formulario que contiene al botón; decimos que su alcance es privado.



## Tipos de botones

Los botones pueden ser:

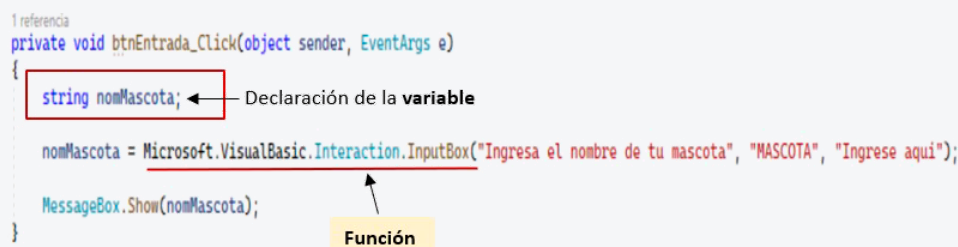
- MessageBoxButtons.Ok
- MessageBoxButtons.YesNo
- MessageBoxButtons.OkCancel
- MessageBoxButtons.RetryCancel
- MessageBoxButtons.YesNoCancel
- MessageBoxButtons.AbortRetryIgnore

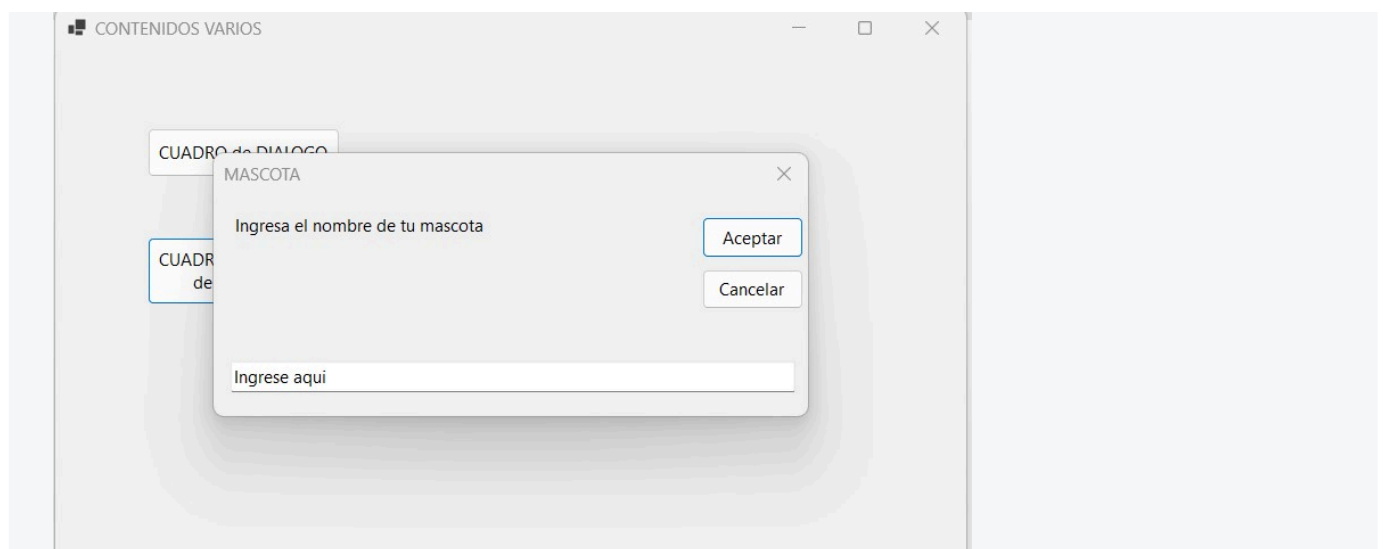
Cuando el cuadro de diálogo tiene más de un botón, el resultado de la función se almacena en una variable para evaluar en las líneas de código que camino hay que tomar.

También existen cuadros de diálogo que permiten ingresar datos desde teclado. Es un cuadro que contiene un textBox.

Para recuperar el dato que contiene el cuadro de ingreso se utilizan variables para guardar el resultado del mismo.

Por ejemplo, si queremos pedir el nombre de una mascota podemos hacerlo de la siguiente manera:





Si se cliquea en **Aceptar** se almacena en la variable lo que escribimos, pero si se cliquea **Cancelar** esa variable queda vacía.



## Cómo recuperar datos de un TextBox

Cuando se ingresan datos en un control TextBox el contenido se guarda en la propiedad **text**

Para recuperar esa información simplemente se lo asignamos a una variable u objeto.

¿Cómo? Se escribe el nombre del objeto luego un punto y después la propiedad.

Suponemos el siguiente diseño:

El TextBox se llama **txtNomMas**

Nombre de Mascota

VER

```
1 referencia
private void btnVer_Click(object sender, EventArgs e)
{
    string entrada;
    entrada = txtNomMas.Text;

    MessageBox.Show(entrada);
}
```

En la variable entrada se guarda el ingreso, observá que el dato del textBox siempre es un **string**

El MessageBox es simplemente para verificar si es correcto.