# Propiedades/Atributos de controles visuales en c#

Las propiedades o atributos de un objeto o componente definen las características del mismo, como el tamaño color o ubicación en la pantalla, o un aspecto de su comportamiento, por ejemplo, si está habilitado o si es o no visible para el usuario. Para cambiar las características de un objeto se tiene que cambiar la propiedad correspondiente. A continuación, una lista de las propiedades más comunes y utilizadas:

**Text**: esta propiedad está presente en muchos los objetos visuales, sobre todos los que contienen algún texto para mostrar, cómo ser, TextBox, Botones formularios, etc. En ella indicamos el texto que aparecerá en el control. Esta propiedad puede ser cambiada tanto en tiempo de ejecución como en tiempo de diseño. También podemos tomar el texto que haya introducido el usuario para utilizarlo durante la ejecución de la aplicación.

**Font**: Sirve para indicar el tipo y tamaño de letra y la poseen los controles que tienen la propiedad Text.

**TextAlign**: esta propiedad define el tipo de alineación que tendrá el texto dentro del control, y puede ser izquierda centrada o derecha.

**Name**: esta propiedad la tienen todos los controles visuales. Esta propiedad no puede estar vacía por lo cual al crear un objeto nuevo se le asigna un “name” por defecto, por ejemplo, si colocamos una caja de texto (textbox) el nombre que tendrá por defecto será “textbox1”. Este será el nombre con el que se conocerá el control cuando lo utilicemos desde el código. En un mismo formulario no puede haber 2 controles con el mismo nombre. Siempre es conveniente cambiar el nombre por defecto y utilizar un nombre que represente la función que tiene el controle en la aplicación para que el código quede más claro. Ejemplo, si en el control de textbox que agregamos se va a guardar la dirección de una persona podemos asignarle el nombre de “txtDireccion”

**Enabled**: Esta es una propiedad booleana y por ende puede tomar los valores verdadero o falso. Esta propiedad indica si el control está o no habilitado para su uso. Por defecto el valor de esta propiedad es verdadero.

Button1.Enabled = False;

**Visible**: esta es una propiedad que sirve, al igual que la propiedad Enabled, e indica si el control está o no visible en tiempo de ejecución. Para cambiar su valor se hace de la siguiente forma

Button1.Visible = false;

**Size:** Esta propiedad es fundamental en la programación de interfaces gráficas en C#, ya que define las dimensiones de un control visual. establece el ancho y el alto de un control. Es una estructura que contiene dos valores:

**Height**: esta es una propiedad que está presente en todos los objetos visuales y nos indica el alto del objeto, utilizando los pixeles como unidad de medida. Para cambiar su valor se hace de la siguiente forma:

Button1.Height = 100

**Width:** Esta propiedad sirve, al igual que la de Height para determinar la dimensión de un objeto visual, indicando el largo del mismo. Para cambiar su valor se hace de la siguiente forma:

Button1.Width = 50

**Location**: Esta propiedad también está presente en todos los controles y es la que determina la posición de dicho control dentro de un formulario o pantalla. Location es una propiedad compuesta por 2 atributos “X” e “Y”, las cuales hacen referencia a la posición del control en el formulario

Button1.Location.X = 16

Button1.Location.Y= 100

**BackColor:** La propiedad BackColor establece el color de fondo de un control. Puedes establecer esta propiedad para cambiar el color de fondo de diversos controles, como botones, etiquetas, paneles, etc. Cambiar el color de fondo puede afectar cómo se ve el control en diferentes temas y estilos del sistema operativo. Si no se establece, el control utiliza su color de fondo predeterminado. Algunos controles pueden soportar colores transparentes.

**ForeColor:** la propiedad ForeColor establece el color del texto o del contenido principal del control. Se utiliza para cambiar el color del texto en controles como botones, etiquetas, cuadros de texto, etc. Siempre es importante elegir un color de texto que contraste bien con el color de fondo (BackColor) para asegurar la legibilidad. Algunos controles pueden no respetar completamente los colores si están configurados para seguir los estilos visuales del sistema operativo.

**Dock:** La propiedad Dock determina cómo un control se acopla a los bordes de su contenedor principal (como un formulario o un panel). se utiliza para alinear un control a uno de los bordes del contenedor o para hacer que llene el área disponible.

DockStyle.None: No se acopla (valor predeterminado).

DockStyle.Top: Acopla el control al borde superior del contenedor.

DockStyle.Bottom: Acopla el control al borde inferior del contenedor.

DockStyle.Left: Acopla el control al borde izquierdo del contenedor.

DockStyle.Right: Acopla el control al borde derecho del contenedor.

DockStyle.Fill: El control llena todo el espacio disponible del contenedor.

**TabIndex:** La propiedad TabIndex en C# es una propiedad crucial para la gestión del orden de navegación por teclado en los controles de una interfaz de usuario en aplicaciones de Windows Forms, esta determina el orden en el que los controles reciben el foco cuando el usuario navega a través de ellos utilizando la tecla Tab. Para establecer el orden de tabulación, hay que asignar un valor entero a la propiedad TabIndex de cada control. Los valores más bajos reciben el foco antes que los valores más altos. Solo los controles que están habilitados (Enabled = true) y visibles (Visible = true) participan en el orden de tabulación. Los controles deshabilitados o invisibles se omiten.

**📝 Ejercitación**

Crear una aplicación que tenga un label y un botón, al presionar el botón el label deberá ubicarse en una posición aleatoria en el formulario. El label debera ir incrementando su valor cada vez que el usuario hace click en el botón, comenzando por cero