

Algunos tics para VS

Solo Números en los TextBox C# - Windows Forms

El siguiente código, se agrega al evento **KeyPress** de la caja de texto, ejemplo:

```
private void txtCaracter_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(char.IsNumber(e.KeyChar)) && (e.KeyChar != (char)Keys.Back))
    {
        MessageBox.Show("Solo se permiten números", "Advertencia",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        e.Handled = true;
        return;
    }
}
```

El código anterior permite además, digitar la tecla *BackSpace* como lo indica la siguiente la línea:
e.KeyChar != (char)Keys.Back)

<http://devcate.blogspot.com/2012/09/solo-numeros-en-los-textbox-windows.html>

Solo Letras en los TextBox C# - Windows Forms

El siguiente código, se agrega al evento **KeyPress** de la caja de texto, ejemplo:

```
private void txtCaracter_KeyPress(object sender, KeyPressEventArgs e)
{
    if (!(char.IsLetter(e.KeyChar)) && (e.KeyChar != (char)Keys.Back))
    {
        MessageBox.Show("Solo se permiten letras", "Advertencia",
            MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        e.Handled = true;
        return;
    }
}
```

El código anterior permite además, digitar la tecla *BackSpace* como lo indica la siguiente la línea:
e.KeyChar != (char)Keys.Back)

<http://devcate.blogspot.com/2012/09/solo-letras-en-los-textbox-windows-forms.html>

Enviar correo electrónico con C# en ASP.NET

Mirar este link:

<http://devcate.blogspot.com/2012/09/codigo-enviar-correo-con-aspnet.html>

Crear Setup o Instalador en Visual Studio .NET 2010

Mirar este link:

<http://devcate.blogspot.com/2012/11/crear-setup-o-instalador-en-visual.html>

Función IsNumeric .Net C#

La **función IsNumeric en C#** nos permite evaluar si una expresión es numérica o no, esto lo podemos lograr a través del siguiente ejemplo:

```
public bool IsNumeric(object Expression)
{
    bool isNum;
    double retNum;
    isNum = Double.TryParse(Convert.ToString(Expression),
        System.Globalization.NumberStyles.Any,
        System.Globalization.NumberFormatInfo.InvariantInfo, out retNum);
    return isNum;
}
```

Random: Generar números aleatorios con C#

Una forma sencilla de **generar números aleatorios** en .Net C# es a través del namespace *System.Random*. Esta clase se puede trabajar a través de tres métodos, pero primero vamos a instanciar la clase **random**:

```
//Creamos un objeto Random, simplemente
Random r = new Random();
// ó podríamos agregar un agregado de aleatoriedad de la siguiente forma:
Random r = new Random(DateTime.Now.Millisecond);
```

Luego de crear el objeto, **conozcamos los tres métodos** que nos permiten generar números aleatorios:

1. Método Next

```
int nAleatorio1 = r.Next();

// Establecemos un máximo
int nAleatorio2 = r.Next(100);

// Establecemos un rango, por ejemplo queremos un número de 3 cifras
int nAleatorio3 = r.Next(100, 999);

// Si deseamos un número entre el 5 y el 15,
int nAleatorio4 = r.Next(5, 15);
```

2. Método NextDouble

```
// Devuelve un double entre 0 y 1
int nAleatorio5 = r.NextDouble();
```

3. Método NextBytes

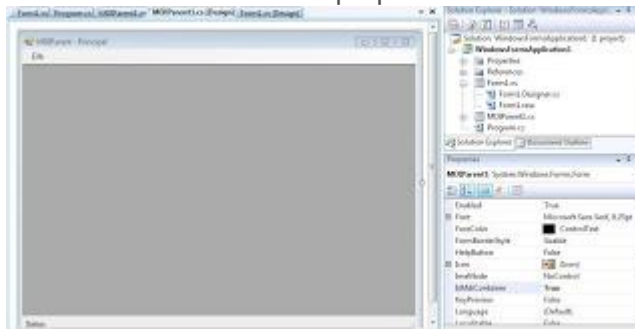
```
byte[] buffer = new byte[100];
r.NextBytes(buffer)
```

Veamos un ejemplo sencillo, donde generamos un **número aleatorio** de 8 cifras, este ejemplo retorna una cadena con el número aleatorio.

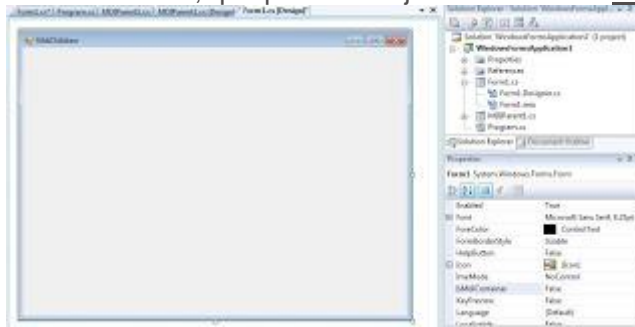
```
public System.String GenerateRandom()
{
    System.Random randomGenerate = new System.Random();
    System.String sPassword = "";
    sPassword = System.Convert.ToString(randomGenerate.Next(00000001, 99999999));
    return sPassword.Substring(sPassword.Length - 8, 8);
}
```

Formulario MDI Child C# en Visual Studio 2012

Los **Formularios MDI** son aquellos que nos permiten contener otros formularios dentro de ellos. A continuación vamos a trabajar un ejemplo en C# Visual Studio 2012, que cual tendrá un formulario principal - MDIParent , a éste se le debe establecer la propiedad *IsMdiContainer* en "true".



Luego nos creamos nuestro formulario Mdi, que para este ejercicio se llama **Form1** - MdiChildren.



Para llamar nuestro formulario MdiChildren una sola vez, desde el formulario principal, podemos crear el siguiente método y llamarlo en el evento Click del menú, así:

```
private void CallFormMdi ()
{
    bool encontrado = false;
    foreach (Form form in this.MdiChildren)
    {
        if (form.Name.Equals ("Form1"))
        {
            encontrado = true;
            form.Activate();
        }
    }

    if (!encontrado)
    {
        Form1 _frmGenerarArchivo = new Form1();
        _frmGenerarArchivo.MdiParent = this;
        _frmGenerarArchivo.Show();
    }
}

private void OpenForm(object sender, EventArgs e)
{
    CallFormMdi();
}
```