



el Guille, la Web del Visual Basic, C#, .NET y más...

[Lo+](#) - [WinFX](#) - [.NET](#) - [ADO.NET](#) - [ASP.NET](#) - [Cómo...](#) - [Colabora](#) - [VB6](#) - [API](#) - [HTML](#) - [Vista](#) - [Links](#) - [Foros](#)



Enviar correo con ASP.NET 2.0

Publicado el 31/Ene/2006
Actualizado el 23/May/2007
Autor: Guillermo 'guille' Som

En este artículo te explico como enviar un correo desde una pagina de ASP.NET (.aspx) usando las clases de System.Net.Mail que son nuevas en la versión 2.0 de .NET Framework.

[Crear una Página Web](#) Diseña tu Sitio Web en Flash Editor de Web Rápido Fácil y Gratis www.Wix.com/Constructor-Sitios-1

[Intercity Exchange](#) correo exchange e-mail microsoft outlook mail www.intercity.cl/exchange.asp

[Promoción Oferta del día](#) Todo los días una súper Oferta Regístrate ahora y recibe tu cupón www.GROUPON.cl

Anuncios Google

Enviar correo desde una página ASPX usando ASP.NET v2.0

En este artículo vamos a ver cómo enviar correos desde páginas de ASP.NET (.aspx), usando las clases del espacio de nombres **System.Net.Mail**, que son nuevas para la versión 2.0 de .NET Framework.

De ese espacio de nombres usaremos tres clases: **MailMessage** que será la clase con la que crearemos el mensaje, y en la que indicaremos el que envía el mensaje, quién lo recibe, el asunto, etc. Para indicar el que envía el mensaje, tendremos que usar un objeto del tipo **MailAddress**, el cual lo asignaremos a la propiedad **From** de la clase **MailMessage**. Para enviar el mensaje usaremos un objeto del tipo **SmtpClient**, que a diferencia de ASP.NET 1.x no tiene métodos estáticos (compartidos), por tanto tendremos que crear una nueva instancia de esa clase y asignar los valores apropiados para poder mandar el mensaje, ahora veremos como hacerlo, además de ver otras cosas que en el equivalente de la versión 1.x no existen.

Estas clases solamente se pueden usar con la versión 2.0 de .NET Framework, si te interesa saber cómo enviar correos desde una aplicación creada con la versión 1.x de ASP.NET, mira este otro artículo: [Enviar correo con ASP.NET v1.x](#)

Y si quieres ver el código para enviar correo desde una páginas ASP (no de .NET), puedes ver este otro artículo: [Enviar correo desde una página ASP](#).

Crear el mensaje

Lo primero que debemos hacer es crear el mensaje que vamos a enviar. Como te he comentado antes, para crear ese mensaje usaremos la clase **MailMessage** del espacio de nombres

System.Net.Mail, a la que le indicaremos los datos habituales:

- El que envía el mensaje (propiedad From)
- A quién va dirigido el mensaje (propiedad To)
- El asunto del mensaje (propiedad Subject)
- El cuerpo (o contenido) del mensaje, (propiedad Body)
- Y algunas cosas más como el formato del mensaje y la prioridad.

Aunque como veremos, la forma de hacerlo difiere "un poco" a como se hacía en las versiones anteriores, al menos en lo que se refiere a los valores de las propiedades From y To.

En el código que te muestro a continuación, se supone que tenemos un formulario Web con los controles (de tipo TextBox) para cada una de esas propiedades.

Lo primero es crear el objeto del tipo **MailMessage**:

```
Dim correo As New System.Net.Mail.MailMessage
```

```
System.Net.Mail.MailMessage correo = new System.Net.Mail.MailMessage();
```

A continuación asignamos las propiedades usando los valores de las cajas de texto que tendrá nuestro WebForm (formulario Web). Como comprobarás, para asignar el valor del correo que envía el mensaje, usamos un objeto del tipo **MailAddress**, ya que la propiedad **From** es de ese tipo y por tanto no acepta una cadena de texto. En el caso de la cuenta **To**, es decir, a quién va dirigido el mensaje, en realidad es una colección de tipo **MailAddressCollection**, por lo que podemos deducir que en realidad se pueden asignar varias direcciones de correo, tantas como necesitemos indicar para mandar el mismo mensaje a todas ellas, pero no te asustes, como puedes ver a continuación es fácil asignar esas cuentas de correo.

Veamos el código para Visual Basic y para C#:

```
correo.From = New System.Net.Mail.MailAddress(txtDe.Text)
correo.To.Add(txtPara.Text)
correo.Subject = txtAsunto.Text
correo.Body = txtTexto.Text
correo.IsBodyHtml = False
correo.Priority = System.Net.Mail.MailPriority.Normal
```

```
correo.From = new System.Net.Mail.MailAddress(txtDe.Text);
correo.To.Add(txtPara.Text);
correo.Subject = txtAsunto.Text;
correo.Body = txtTexto.Text;
correo.IsBodyHtml = false;
correo.Priority = System.Net.Mail.MailPriority.Normal;
```

Como puedes ver, el que envía el mensaje, (propiedad **From**), hay que indicarlo asignándole un objeto del tipo **MailAddress**, en este caso, lo creamos directamente y ese nuevo objeto es el que asignamos a la propiedad **From**.

También te comentaba que el o los destinatarios había que asignarlo a la propiedad **To**, que es una colección del tipo **MailAddressCollection**, y como en cualquier colección, cada nuevo elemento lo añadimos usando el método **Add** de la colección, en este caso, los elementos son cadenas con la dirección de correo de los destinatarios. En nuestro caso, solo hay uno, el que esté indicado en la caja de textos **txtPara**.

Una vez que ya tenemos el objeto del mensaje creado, tenemos que enviarlo, para ello usaremos una instancia de la clase **SmtpClient** de la que usaremos tres propiedades: **Host** a la que le indicaremos el nombre del servidor de correo por el que queremos enviar el mensaje, este servidor de correo suele toma la forma de **mail.servidor.com** o **correo.servidor.com**, dependiendo de la versión o idioma en el que esté configurado el host del servidor de correo.

```
Dim smtp As New System.Net.Mail.SmtpClient
smtp.Host = "servidor de correo"
```

```
System.Net.Mail.SmtpClient smtp = new System.Net.Mail.SmtpClient();
smtp.Host = "servidor de correo";
```

Si el servidor de correo necesita autenticación, podemos hacerlo mediante un objeto del tipo **NetworkCredential**, en el que indicaremos el nombre del usuario y la clave, ese objeto lo asignaremos a la propiedad **Credentials** del objeto **SmtpClient** que acabamos de crear.

```
smtp.Credentials = New System.Net.NetworkCredential("usuario", "password")
```

```
smtp.Credentials = new System.Net.NetworkCredential("usuario", "password");
```

Por último, nos queda la parte más importante: enviar el mensaje. Para ello usaremos el método **Send** del objeto creado a partir de la clase **SmtpClient**, al que le pasaremos el objeto de tipo **MailMessage** que hemos creado al principio.

El uso de este método lo ponemos dentro de un Try/Catch ya que puede producirse un error al enviar ese mensaje, por ejemplo, si el servidor de correo requiere autenticación o que no se ha podido enviar el mensaje por la causa que sea.

El mensaje de error o acierto lo mostraremos en una etiqueta llamada **LabelError**.

```
Try
    smtp.Send(correo)
    LabelError.Text = "Mensaje enviado satisfactoriamente"
Catch ex As Exception
    LabelError.Text = "ERROR: " & ex.Message
End Try
```

```
try
{
    smtp.Send(correo);
    LabelError.Text = "Mensaje enviado satisfactoriamente";
}
catch(Exception ex)
{
    LabelError.Text = "ERROR: " + ex.Message;
}
```

Y esto es todo...

Nos vemos.
Guillermo

P.S.
23/Dic/2006:

El código de las páginas de envío:

[Código de Visual Basic](#)

[Código de Visual C#](#)

Para que veas cómo queda el código anterior: [Enviar correo usando Visual Basic](#).

Si al pulsar en este link te muestra un error es que no tengo activada la versión 2.0 de ASP.NET en este sitio, a la hora de escribir esto (23/Dic/2006) en elGuille.info tengo la versión 2.0 y en MundoProgramacion.com tengo la 1.1, aunque ya está (o debería estar) la versión 2.0 en los dos sitios.

En cualquier caso, esa página no es "operativa", es decir, no envía correos, aunque si te sirve para ver cómo quedaría la página que te muestro en los dos links anteriores.

Nota:

Debes tener en cuenta que tendrás que rellenar con tus datos los valores del servidor y si necesitas autenticación, debes indicar los datos del usuario y la contraseña.

