

Podemos pasar información entre páginas de varias maneras, algunas de las cuales dependen de cómo se produzca la redirección.

Las siguientes opciones están disponibles incluso si la página de origen está en una aplicación web ASP.NET diferente de la página de destino, o si la página de origen no es una página web ASP.NET:

- Utilizar una cadena de consulta; propiedad QueryString del objeto HttpRequest (*trabajada en la tarea anterior*)
- Obtener información HTTP POST de la página de origen.

Las siguientes opciones están disponibles solo cuando las páginas de origen y de destino están en la misma aplicación web ASP.NET.

- Usar estado de sesión (*la trabajaremos más adelante*)
- Obtener la información en la página de destino a partir de los controles de la página de origen (*propiedad PreviousPage*); esta estrategia la usaremos si la página de origen no expone propiedades públicas que contengan la información que necesitamos
- Crear propiedades públicas en la página de origen y acceder a los valores de propiedad en la página de destino (*directiva PreviousPageType*)

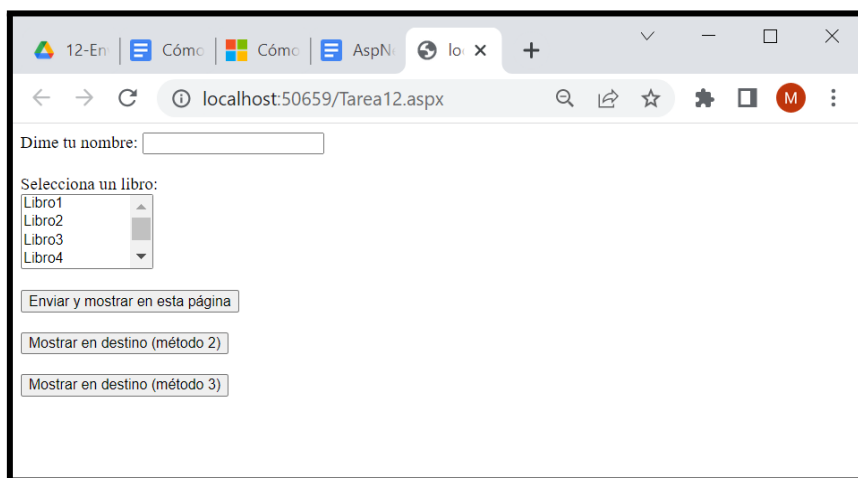
En esta tarea pondrás en práctica las dos últimas opciones de la lista. Ambos métodos son útiles en dos situaciones en las que invocas la página de destino:

- en una devolución de datos entre páginas, que es una transferencia basada en el cliente), y
- con el método Transfer, que es una operación basada en el servidor

En ambas operaciones, el código de la página de destino puede obtener una referencia a la página de origen mediante la propiedad *PreviousPage*.

### Práctica guiada

Añade a SitioWebTuNombre el WebForm **Tarea12-Envío entre páginas-II.aspx** de la imagen



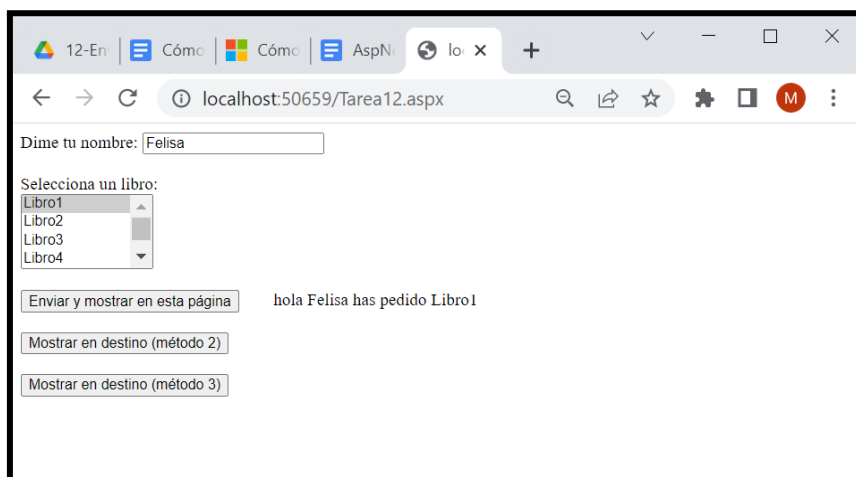
La pulsación del **primer botón** provoca un PostBack solicitando al servidor que muestre en esta misma página la información proporcionada por el usuario. Esto es, una llamada a la misma página, algo muy común en ASP .NET. Lo único que haremos es llenar el texto de la Label en el evento click del botón.

```
Protected Sub Button1_Click(sender As Object, e As System.EventArgs) Handles Button1.Click
```

```
    Label1.Visible = True
```

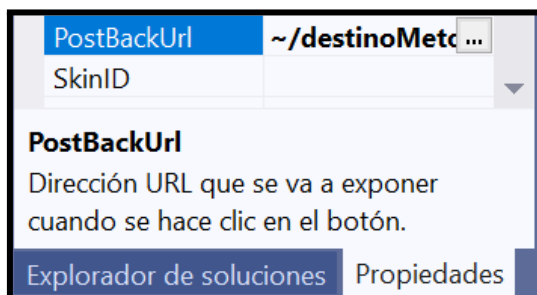
```
    Label1.Text = "hola " & TextBox1.Text & " has pedido " & ListBox1.SelectedItem.ToString()
```

```
End Sub
```



El **segundo botón** ha de mostrar la misma información pero en un segundo webform **Tarea12DestinoMétodo2.aspx**.

Algo importante aquí es que debemos asignar valor a la **propiedadPostBackUrl** del botón para especificar la dirección URL que se va a exponer cuando se hace clic en dicho botón. Pestaña propiedades del segundo botón:



(\*fijate que no es necesario agregar ningún código al evento *OnClick* del botón\*)

En el load de la página destino **Tarea12DestinoMétodo2.aspx.vb** haremos referencia a la página origen mediante la propiedad [PreviousPage](#) y el método [FindControl](#)

```
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
```

```
    Dim cpTextBox1 As TextBox
```

```
    Dim cpListBox1 As ListBox
```

```
    cpTextBox1 = PreviousPage.FindControl("TextBox1")
```

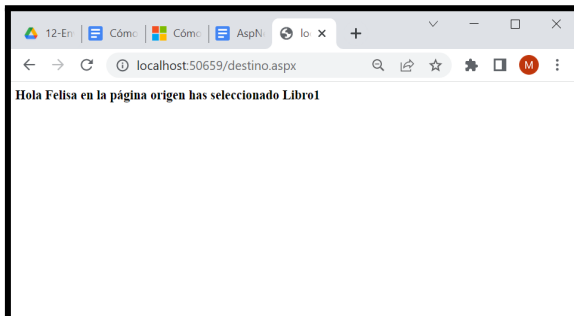
```
    cpListBox1 = PreviousPage.FindControl("ListBox1")
```

```
    Label1.Text = "Hola " & cpTextBox1.Text &
```

```
        " en la página origen has seleccionado " &  
cpListBox1.SelectedItem.ToString
```

```
End Sub
```

Si has seguido las indicaciones debes obtener algo parecido a esto:



El **tercer botón** va a exponer la url de un nuevo webform destino ***Tarea12DestinoMétodo3.aspx***, que al igual que el anterior mostrará la información que el usuario proporciona en la página origen, pero lo hará con el último de los métodos que se han citado al inicio de este documento: crear propiedades públicas en la página de origen y acceder a los valores de propiedad en la página de destino.

Configura la propiedad PostBackUrl del botón (*Tarea12DestinoMétodo3.aspx*)

Añade el siguiente código a continuación de las líneas *Partial Class ...* e *Inherits...* de la página origen (*formulario de inicio de esta tarea*).

```
'Estas propiedades encapsulan los controles
```

```
'para que puedan ser accedidos desde otras páginas.
```

```
Public ReadOnly Property ListaPublica() As ListBox
```

```
Get
```

```
    Return ListBox1
```

```
End Get
```

```
End Property
```

```
Public ReadOnly Property TextoPublico() As TextBox
```

```
Get
```

```
Return Me.TextBox1
```

```
End Get
```

```
End Property
```

Implementa la directiva **PreviousPageType** en la página destino:

```
<%@ Page Language="VB" AutoEventWireup="false" CodeFile="DestinoMetodo3.aspx.vb"
Inherits="DestinoMetodo3" %>
<%@ PreviousPageType VirtualPath="~/Tarea12.aspx" %>
```

Por último añade el siguiente código en la página destino:

```
Protected Sub Page_Load(sender As Object, e As EventArgs) Handles Me.Load
    Label1.Visible = True
    Label1.Text = " Hola " + PreviousPage.TextoPublico.Text + ", en la página
origen has seleccionado: " + PreviousPage.ListaPublica.SelectedValue.ToString()
End Sub
```

*(\*al invocar PreviousPage Visual Studio nos ofrece las propiedades públicas para poder trabajar con ellas\*)*

Si has seguido las indicaciones debes obtener algo parecido a esto:

