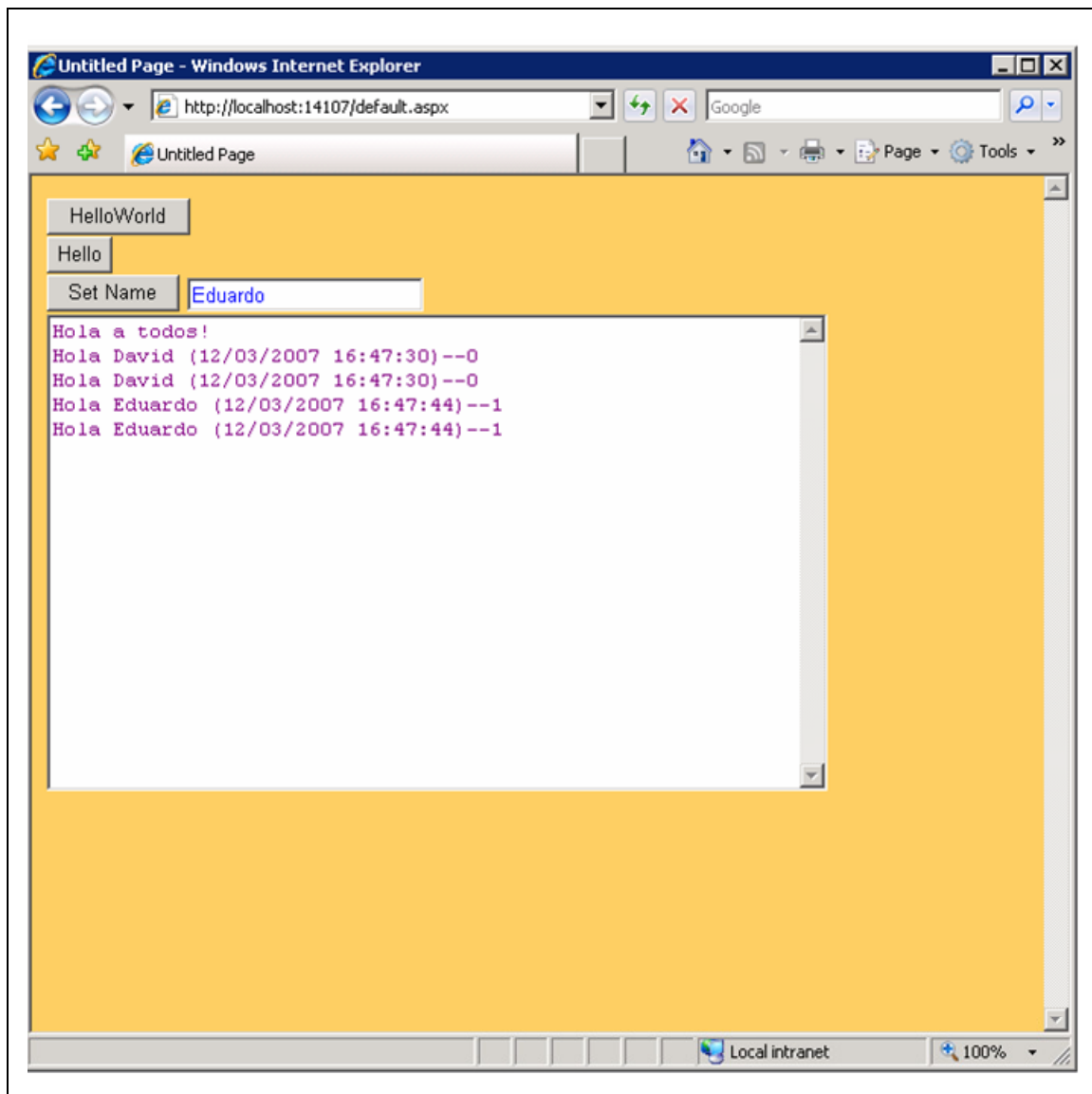


Boletín I de ASP.Net. Conceptos básicos.

El objetivo de este boletín es introducirnos en ASP.Net, para ello el primer paso que vamos a dar es crear una aplicación Web ASP.Net con la misma funcionalidad que cada aplicación Windows desarrollada hasta el momento.

HelloWorldWeb

En esta primera aplicación Web haremos uso del Servicio Web HelloWorldWS, la aplicación tiene el siguiente aspecto:



Como novedad en ella hemos utilizado una hoja de estilos o CSS. La hoja de estilos sirve para definir el estilo (color, tipo de letra, puntero de ratón, etc, ..) de las páginas Web que conforma una aplicación Web corporativa. Por ejemplo:

```
body
{
    font-size: larger;
    font-family: 'Comic Sans MS' , Verdana, Arial;
    background-color: #ffcc66;
}

.in
{
    color: blue;
}

.out
{
    color: purple;
}
```

Define el tipo de letra y el color de fondo de cuerpo de las páginas de nuestra aplicación. Además también define dos clases de estilo (in y out) que después pueden ser utilizadas en controles individuales a través de la propiedad `CssClass`. Para que una página utilice una hoja de estilos sólo es necesario que incluya una referencia a la misma en su cabecera:

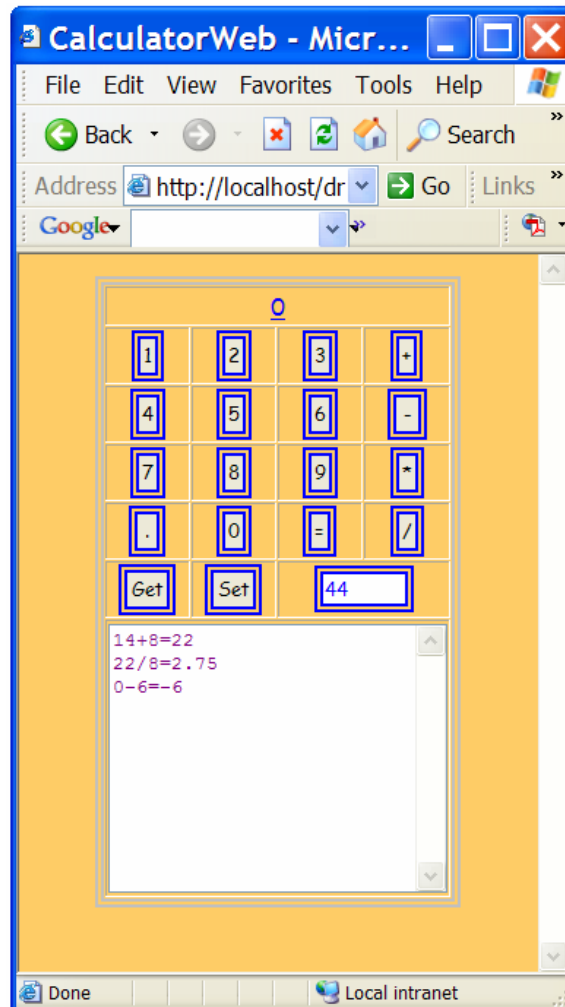
```
<LINK href="./default.css" type="text/css" rel="stylesheet">
```

Otra diferencia importante con respecto a las aplicaciones Windows es que estado de la aplicación sólo se puede guardar haciendo uso de cookies, y, al igual que los servicios Web, para almacenar variables haremos uso de la propiedad `Session`. Por ejemplo, la referencia al Servicio Web sería:

```
protected localhost.HelloWorldWS ws
{
    get
    {
        localhost.HelloWorldWS result;
        if (Session["ws"] != null)
            result = (localhost.HelloWorldWS) Session["ws"];
        else
        {
            result = new localhost.HelloWorldWS();
            result.CookieContainer = new System.Net.CookieContainer();
            Session["ws"] = result;
            resultTxt.Text = "Servicio Web creado con éxito!!\r\n";
        }
        return result;
    }
}
```

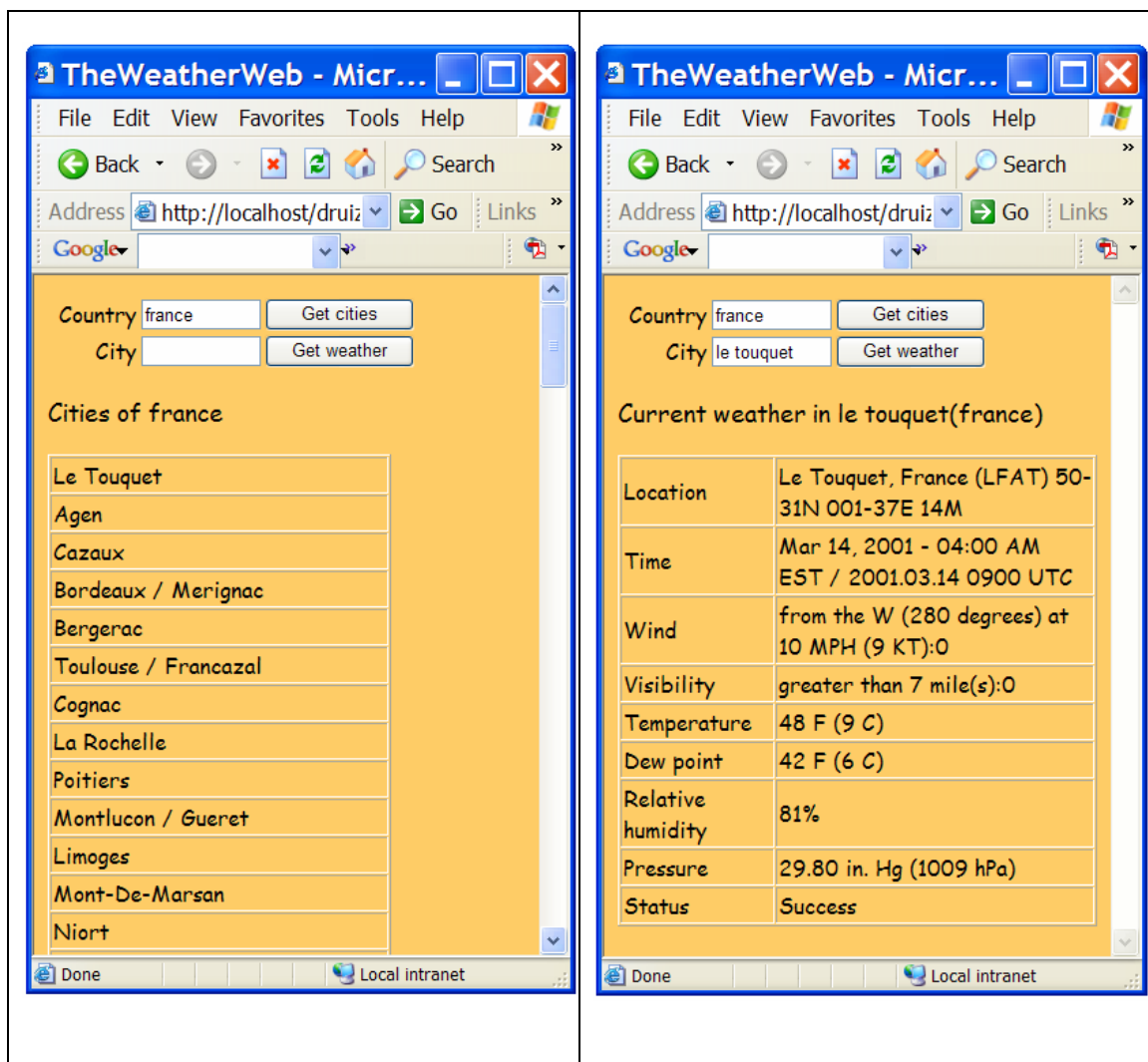
CalculatorWeb

En este ejercicio vamos a practicar con algunos controles Web haciendo una aplicación Web que implemente una calculadora que consuma el Servicio Web CalculatorWeb hecho anteriormente. El aspecto de la aplicación es el siguiente:



TheWeatherWeb

El objetivo de este ejercicio es hacer una aplicación Web que consuma el Servicio Web del tiempo proporcionado por www.webservicesx.net. En este ejercicio vamos a mostrar los resultados en un control XML. A este control en tiempo de ejecución se le puede decir que cargue un archivo XML (o una cadena) y que le de formato según una plantilla de transformación XSLT. La siguiente figura muestra un par de pantallas con la aplicación:



La tabla HTML en la que se muestran los datos es el resultado de aplicar una transformación XSLT a la cadena XML que devuelve el Servicio Web. Por

ejemplo, para dar formato a las ciudades, el patrón de transformación XSLT es el siguiente¹:

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
  <table cellSpacing="1" cellPadding="1" border="1">
    <xsl:for-each select="NewDataSet/Table">
      <tr><td><xsl:value-of select="City"/></td></tr>
    </xsl:for-each>
  </table>
</xsl:template>
</xsl:stylesheet>
```

Para configurar el objeto WebControl.XML basta con dar los valores adecuados a las propiedades TransformSource (con el archivo XSLT que contiene la transformación) y DocumentContent (con la cadena XML a transformar):

```
string result = ws.GetCitiesByCountry(countryTbx.Text);
resultXml.TransformSource = "getCities.xslt";
resultXml.DocumentContent = result;
```

¹ Se sale del objetivo de este curso aprender en profundidad XSLT, un buen sitio donde puede encontrar un tutorial sobre este lenguaje es www.w3schools.com.

LectureNotesASPNetWeb (Sin Master Pages)

El objetivo de este ejercicio es crear una página con marcos (frames) clásico de tipo banner-contents-main. En el marco contents pondremos una formulario ASPX que servirá para cargar en el marco main la aplicación Web correspondiente. Para hacer esto es necesario que utilicemos script, ya que es imposible desde el servidor tener conocimiento sobre los marcos de una página Web. Podemos utilizar el siguiente método:

```
protected void redirect(string page, string frame)
{
    Response.Write("<script>window.open(\"\" + page + \"\", \"\" + frame + \"\");</script>");
}
```

El aspecto del conjunto de marcos es el que se muestra a continuación:

