



SKINS Y THEMES.....	2
Aplicando Estilos a Controles HTML	2
Aplicando Estilos a Controles Web de Servidor.....	3
Usar Temas para Personalizar un Sitio	3
La carpeta App_Themes.....	4
Temas Globales y de Aplicación.....	5
Asignando un Tema a una Página.....	5
Asignando un Tema en Web-Config	5
Deshabilitando Temas para un Control.....	5
Skins identificados en un Tema.....	6
Skins Por Defecto e Identificados	6
Estilos del Lado de Servidor Utilizando Temas.....	7
Contenidos de un Tema y Skin	7
Utilizando CSS en Temas.....	7
Utilizando Imágenes en Temas.....	7
Tematizando Grupos de Controles	8
Asignando un Tema Mediante Código.....	9
Ejercicio:.....	10



SKINS Y THEMES

Temas y Skins - La característica de Temas y Skins de ASP.NET nos permite factorizar la información de estilo y diseño en un grupo separado de ficheros, a cuyo conjunto llamamos "Tema".

Un Tema se puede aplicar a cualquier sitio de forma que afecte a la apariencia y entorno de las páginas y controles del sitio. Los cambios en el Estilo de un sitio pueden administrarse realizando cambios al Tema, sin tener que editar las páginas de forma individual. Los Temas también se pueden compartir con otros desarrolladores.

ASP.NET incluye varias características para personalizar la apariencia o el estilo de las páginas y los controles de nuestra aplicación. Los controles soportan un modelo de objeto Style para establecer las propiedades de estilo como fuentes, bordes, colores de fondo y primer plano, anchura, altura, etc. Los controles también soportan completamente "Cascading Style Sheets" (CSS) para factorizar los ajustes de estilo de forma separada de las propiedades de los controles. Podemos definir la información de los estilos como propiedades de los controles o CSS, o podemos definir esta información en grupos de ficheros separados llamados Theme (Tema), que se aplicarán a todas o una parte de las páginas de la aplicación. Los estilos de controles individuales se especifican como un Skin dentro de un Tema.

Aplicando Estilos a Controles HTML

Las etiquetas estándar HTML soportan CSS a través del atributo "style" (estilo), que se puede fijar a una lista de parejas atributo/valor delimitada por puntos y comas.

CSS también define un atributo de clase que puede fijarse a una definición de estilo CSS contenida en una sección <style>...</style> del documento. El atributo de clase hace más sencillo definir los estilos una vez y aplicarlos a diferentes etiquetas sin tener que redefinir el estilo. Los estilos en controles HTML de servidor también se pueden establecer de esta forma.

Cuando se analiza una página ASP.NET, la información de estilo se rellena dentro de una propiedad Style (del tipo CssStyleCollection) dentro de la clase **System.Web.UI.HtmlControls.HtmlControl**. Esta propiedad es, esencialmente, un diccionario que muestra los estilos del control como grupos de valores indexados por cadenas, para cada clave de "style-attribute".

Por ejemplo, podemos usar el siguiente código para fijar y, consecuentemente, obtener el atributo de estilo "width" (ancho) de un control de servidor HtmlInputText.

```
<script language="C#" runat="server" >
    void Page_Load(Object sender, EventArgs E) {
        MyText.Style["width"] = "90px";
        Response.Write(MyText.Style["width"]);
    }
</script>

<input type="text" id="MyText" runat="server"/>
```



Aplicando Estilos a Controles Web de Servidor

Los controles Web de Servidor proporcionan un nivel adicional de soporte para estilos, añadiendo varias propiedades fuertemente tipadas para los ajustes de estilos más usados, tales como color de fondo, nombre y tamaño de la fuente, etc.

Estas propiedades de estilo representan un subconjunto de los comportamientos de estilo disponibles en HTML y se representan como propiedades llanas accesibles directamente a través de la clase base **System.Web.UI.WebControls.WebControl**. La ventaja de usar estas propiedades es que proporcionan comprobación en tiempo de compilación y finalización de declaraciones en herramientas de desarrollo como Microsoft Visual Studio .NET.

El espacio de nombres System.Web.UI.WebControls incluye una clase base Style que encapsula los atributos de estilo comunes (las clases adicionales de estilo, como TableStyle y TableItemStyle heredan de esta clase base común).

Los estilos de los controles Web de servidor se pueden establecer mediante código utilizando el método **ApplyStyle** de la clase base WebControl, cómo vemos en el siguiente código.

```
using System.Drawing;
<script language="C#" runat="server">
    void Page_Load(Object Src, EventArgs E) {
        Style style = new Style();
        style.BorderColor = Color.Black;
        style.BorderStyle = BorderStyle.Dashed;
        style.BorderWidth = 1;

        MyLogin.ApplyStyle (style);
        MyPassword.ApplyStyle (style);
        MySubmit.ApplyStyle (style);
    }
</script>

Login: <ASP:TextBox id="MyLogin" runat="server" /><p/>
Password: <ASP:TextBox id="MyPassword" TextMode="Password" runat="server" />
View: <ASP:DropDownList id="MySelect" runat="server"> ... </ASP:DropDownList>
```

Usar Temas para Personalizar un Sitio

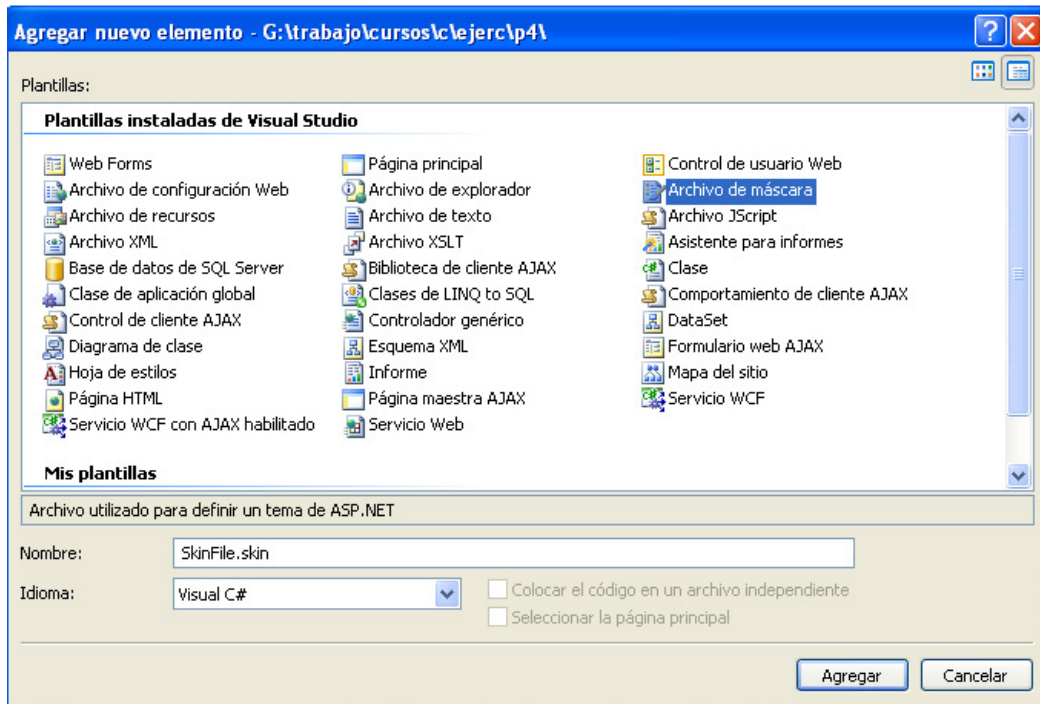
Además de especificar estilos en controles individuales, ASP.NET introduce Temas, los cuales nos ofrecen una manera fácil de definir estilos para controles y páginas de nuestro sitio, de forma separada a las páginas de nuestra aplicación. Así puedes diseñar un sitio sin preocuparte por el estilo y aplicarlo más tarde sin tener que actualizar las páginas o el código. También puedes obtener temas personalizados de una fuente externa para aplicar ajustes estilísticos a tu aplicación.

Otro beneficio es que los ajustes de estilo están almacenados en una única ubicación y pueden ser mantenidos de forma independiente de la aplicación a la que se le aplica el Tema.



El tema aplica automáticamente las propiedades de estilo a los controles en la página en tiempo de ejecución.

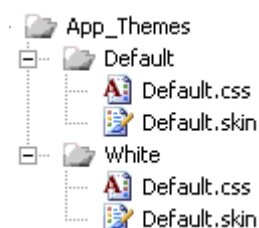
La carpeta App_Themes



Sobre sitio Web botón derecho >> Agregar nuevo elemento (Add new ítem) >> Archivo de máscara (skin file).

Los temas residen en la carpeta especial App_Themes. Un tema consiste en un subdirectorío bajo esta carpeta que contiene un grupo de uno o más ficheros Skin, con extensión .skin. Un tema puede contener también un fichero CSS y/o subdirectorío para ficheros estáticos, por ejemplo imágenes.

La siguiente figura muestra el directorio App_Themes con dos temas definidos, llamados "Default" y "White", cada uno de los cuales tiene un sólo fichero skin y uno CSS.



Un fichero skin son simplemente definiciones de control, **tal y como deben aparecer en la página**. Un fichero de skin puede contener varias definiciones de control, por ejemplo una definición por cada tipo de control.

```
<asp:Label runat="server" CssClass="Label" ForeColor="green">
</asp:Label>
<asp:Label skinid="obl" runat="server" CssClass="Label" BackColor="#d5f8cc">
</asp:Label>

<asp:TextBox SkinID="obl" runat="server" BackColor="#d5f8cc"></asp:TextBox>
```



```
<asp:Image skinid="logo" runat="server"
ImageUrl="~/App_Themes/alternativo/icono_visual_studio.png" />
```

Las propiedades de controles definidas en el tema sobrescriben automáticamente los valores de la propiedad local de un control del mismo tipo en la página en la que aplicamos el tema.

Por ejemplo, una definición de un control del tipo

```
<asp:Calendar Font-Name="Verdana" runat="server"/>
```

que se encuentre en un fichero skin causará que todos los controles Calendar en páginas con el Tema aplicado utilizarán la fuente Verdana.

Es un error especificar un valor de ID de propiedad para una definición de control en un fichero de skin. El estilo aplicado por programación sobrescribirá el de skin.

Temas Globales y de Aplicación

Un tema puede residir en un nivel de aplicación o en un nivel máquina (accesible de forma global para todas las aplicaciones). Los temas de nivel de aplicación se encuentran en el directorio App_Themes bajo el directorio raíz de la aplicación, como se ha descrito anteriormente.

Los temas globales se colocan en el directorio "Themes", bajo una carpeta ASP.NETClientFiles, bajo el directorio de instalación de ASP.NET, por ejemplo %WINDIR%\Microsoft.NET\Framework\<version>\ASP.NETClientFiles\Themes. La ubicación de los temas globales para los sitios web de IIS es Inetpub\wwwroot\aspnet_client\system_web\<version>\Themes.

Asignando un Tema a una Página

Se puede asignar un tema a una página individual estableciendo la directiva <%@ Page Theme="..." %> al nombre del Tema de nivel global o de aplicación (el nombre de la carpeta bajo el directorio Themes o App_Themes). Una página puede tener sólo un tema aplicado, pero en dicho tema pueden haber varios ficheros skin para aplicar configuraciones de estilo a los controles de la página.

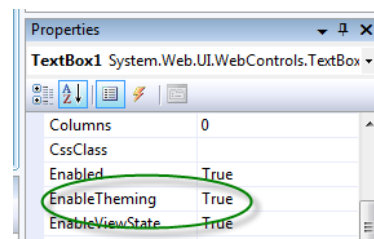
Asignando un Tema en Web-Config

También podemos definir el tema aplicado para todas las páginas de la aplicación especificándolo en la sección <pages theme="..." /> de Web-config. Para quitar este tema en una página particular podemos establecer el atributo Theme de la directiva de la página a un string vacío ("").

Una master page no puede tener un tema aplicado.

Deshabilitando Temas para un Control

Un control específico puede ser excluido de tener que sobrescribir sus propiedades mediante el establecimiento de la propiedad **EnableTheming** a false.





Skins identificados en un Tema

Por defecto, la definición de un control en un fichero de skin se aplica a todos los controles del mismo tipo en las páginas de la aplicación. Sin embargo, podemos querer que los controles del mismo tipo tengan apariencias diferentes en distintas partes de la aplicación.

Por ejemplo, podríamos querer que el texto de un control Label aparezca en negrita en una parte y en italic en otra. Podemos hacer esto mediante **skins identificados** en un tema.

Skins Por Defecto e Identificados

En las definiciones de los controles podemos establecer diferentes propiedades SkinID a nombres de nuestra elección, y después establecer este mismo SkinID en los controles de las páginas que tendrán dicho skin aplicado. En ausencia de la propiedad SkinID, se aplica el skin por defecto (uno sin la propiedad SkinID fijada).

```
<%@ Page Language="C#" Theme="OrangeTheme2" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <title>Named Skins</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h3>Named Skins</h3>
      <asp:Label ID="Label1" runat="server" Text="Hello1" /><br />
      <asp:Label ID="Label2" runat="server" Text="Hello2" SkinID="Blue" /><br>
      <asp:Label ID="Label3" runat="server" Text="Hello3" /><br />
    </div>
  </form>
</body>
</html>
```

Los skins identificados se pueden organizar de diferentes formas dentro de los ficheros de skin de los temas. Debido a que un tema puede contener varios ficheros de skin debemos dividir los skins identificados en diferentes ficheros, de forma que cada uno de estos ficheros contenga las definiciones de controles con el mismo SkinID.

3 ficheros de skin en un Tema, cada uno nombrado según el valor de SkinID:	Según el tipo de control	O por áreas de nuestro sitio
/WebSite1 /App_Themes /MyTheme Default.skin Red.skin Blue.skin	/WebSite1 /App_Themes /MyTheme GridView.skin Calendar.skin Label.skin	/WebSite1 /App_Themes /MyTheme HomePage.skin DataReports.skin Forums.skin



Estilos del Lado de Servidor Utilizando Temas

Un Tema está destinado a ser aplicado a una aplicación después de haberla creado, incluso después de que el sitio ya esté hospedado en un servidor. La persona que aplica el tema a la aplicación no tiene porque ser la misma que la desarrolló.

Cuando se aplica un Tema a una aplicación, las propiedades de estilo que se encuentran en la definición del Tema sobrescriben los valores de los controles de destino en las páginas de aplicación.

Por otro lado, es algo común entre los desarrolladores factorizar información estilística y el comportamiento de servidor por separado. Un ejemplo de esto es la utilización de CSS para definir los estilos de control y de marcado en un fichero separado. También podemos utilizar Temas para este propósito, aplicando el Tema como una especie de hoja de estilos del lado de servidor.

Contenidos de un Tema y Skin

Utilizando CSS en Temas

Una Hoja de Estilos en Cascada (CSS) puede ser añadido a un Tema colocándolo bajo el subdirectorio del Tema. La hoja de estilos CSS se aplicará a todas las páginas a las que apliquemos el tema, siempre que la página tenga un control **<head runat="server"/>** definido. Podemos nombrar el fichero CSS de cualquier forma siempre que tenga la extensión .css.

Un Tema puede contener múltiples ficheros CSS. Cuando la página contiene referencias a los ficheros CSS (mediante una etiqueta **<link rel="stylesheet" href="...">** en el elemento **<head/>**), cualquier fichero CSS del Tema se aplicara **después** de las hojas de estilo de la página.

Utilizando Imágenes en Temas

Un tema también puede contener imágenes que se referencian en la definición del control en el fichero de skin. Las referencias a las imágenes en skins deben utilizar rutas relativas a la carpeta de las imágenes dentro del directorio Theme, de forma que las imágenes y el fichero skin se pueden mover juntas a otras aplicaciones de forma sencilla.

En tiempo de ejecución, las rutas a las imágenes se revisan, de forma que **las referencias son relativas al control dentro de la página objetivo**, en lugar de al fichero skin.



Tematizando Grupos de Controles

La mayoría de las propiedades que establecemos en un skin serán propiedades de valor simple, como por ejemplo Font-Name, Width y BackColor. Sin embargo, también podemos crear un skin de propiedades de grupo.

Haciendo esto no aplicamos propiedades a los elementos en el grupo del control objetivo. En lugar de eso, reemplazamos el grupo en su totalidad al utilizar Temas y combinamos el grupo cuando utilizamos StyleSheetThemes (aquí viene nombre de skin igual que con theme, solo que se carga en primer lugar y se sobrescribe con las propiedades del control de la página que a su vez se sobrescriben con los de theme).

El siguiente ejemplo muestra el grupo Items de un control DropDownList definido en un fichero skin.

```
<asp:RadioButtonList runat="server">
  <asp:ListItem value="5">Option 5</asp:ListItem>
  <asp:ListItem value="6">Option 6</asp:ListItem>
  <asp:ListItem value="7">Option 7</asp:ListItem>
  <asp:ListItem value="8">Option 8</asp:ListItem>
</asp:RadioButtonList>
```

```
<%@ Page Language="C#" Theme="MyTheme" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<script runat="server">
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
  <title>Colección de temas</title>
</head>
<body>
  <form id="form1" runat="server">
    <div>
      <h3> Colección de temas </h3>
      <asp:RadioButtonList ID="RadioButtonList1" runat="server">
        <asp:ListItem Value="1">Option 1</asp:ListItem>
        <asp:ListItem Value="2">Option 2</asp:ListItem>
        <asp:ListItem Value="3">Option 3</asp:ListItem>
        <asp:ListItem Value="4">Option 4</asp:ListItem>
      </asp:RadioButtonList>
    </div>
  </form>
</body>
</html>
```

El resultado una lista desplegable de 8 opciones.

Ésto también es muy útil para ciertas propiedades de grupos que contienen grupos de estilos, como la propiedad LevelStyle del control TreeView o las propiedades LevelMenuItemStyles, LevelSubMenuItemStyles o LevelSelectedStyles del control Menu.



Asignando un Tema Mediante Código

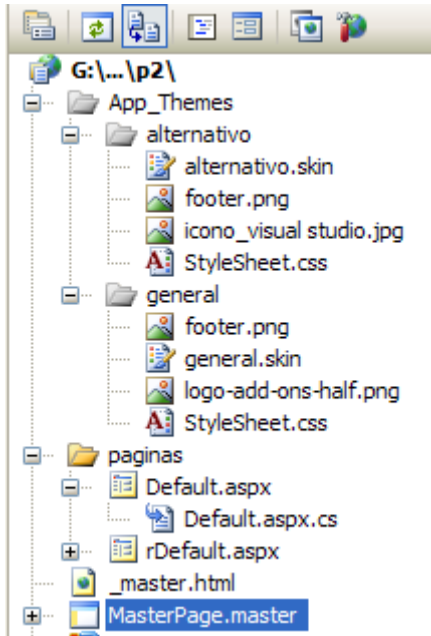
Un tema debe ser aplicado a la página muy pronto en el ciclo de vida de una petición, en el evento PreInit.

```
<script runat="server">
    Protected void Page_PreInit () {
        Page.Theme = Server.HtmlEncode(Request.QueryString("Theme"))
    }
</script>
<html xmlns="http://www.w3.org/1999/xhtml">
<head id="Head1" runat="server">
    <title>Assigning a Theme in Code</title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h2>Assigning a Theme in Code</h2>
            <a href="programmatic_vb.aspx?Theme=ThemeA">Theme A</a>
            <a href="programmatic_vb.aspx?Theme=ThemeB">Theme B</a>
        </div>
    </form>
</body>
</html>
```



Ejercicio:

Siguiendo con el ejercicio anterior: diseña dos skins.



Skin general:

- (label) Lleva una definición para Label normal y otra para Label con un SkinId llamado "obl" que llevará un color de texto azul para los campos obligatorios.
- (textbox) Lleva una definición para campos de texto normal y otra con un fondo amarillo para SkinId "obl".
- (DropDownList) Lleva una definición para listas desplegables normal y otra con fondo amarillo para SkinId "obl".
- (img) Lleva una definición para imágenes de servidor con un SkinId "logo" para la imagen del logotipo.

RECOGIDA DE DATOS

Nombre

Regalo

Fecha entrega

Fecha entrega alternativa

D.N.I.

Indicar sexo y condición

☐ Hombre Funcionario

☐ Mujer funcionaria

☐ Otros

En caso de marcar otros señale que otros

Curso de programación c# en .NET

En este ejemplo vamos a crear una página web que recoja información sobre personas pertenecientes a la comunidad universitaria. Después se hará sorteo para regalar unos bolis.



Skin alternativo:

- (label) Lleva una definición para Label normal y otra para Label con un SkinId llamado "obl" que llevará un color de texto verde para los campos obligatorios.
- (textbox) Lleva una definición para campos de texto normal y otra con un fondo #d5f8cc para SkinId "obl".



- (DropDownList) Lleva una definición para listas desplegables normal y otra con fondo #d5f8cc para SkinId "obl".
- (img) Lleva una definición para imágenes de servidor con un SkinId "logo" para la imagen del logotipo. La imagen del fondo del pie está cambiada.

RECOGIDA DE DATOS

Nombre

Regalo

Fecha entrega

Fecha entrega alternativa

D.N.I.

Indicar sexo y condición

☐ Hombre Funcionario

☐ Mujer funcionaria

☐ Otros

En caso de marcar otros señale que otros

Label



Curso de programación c# en .NET

En este ejemplo vamos a crear una página web que recoja información sobre personas pertenecientes a la comunidad universitaria. Después se hará sorteo para regalar unos bolis.