 Instituto Superior Santo Domingo	<u>UNIDAD Nº :3</u> Recursos de ASP.Net.	<u>TEMAS:</u> Cookies y Session.	Clase 9
--	---	---	------------------------------

Objetivos:

- Creación de Cookies, variables de sesión y aplicación.
- Recuperación de Cookies, variables de sesión.
- Usos de estas herramientas.

Introducción

Cuando un sitio web necesita identificar un usuario que visita un conjunto de páginas web puede emplear cookies y variables de sesión.

Como veremos las cookies se almacenan en el cliente (navegador) y son enviadas al servidor cada vez que le solicitamos una página a dicho servidor. En cambio las variables de sesión se almacenan en el servidor.

Cookies

El protocolo http es un protocolo desconectado. El protocolo http por si mismo no provee al servidor Web si dos peticiones provienen del mismo navegador. Desde este punto de vista, para el Servidor Web cada petición de página proviene de un nuevo usuario. Esta característica puede ser un inconveniente para ciertos sitios Web.

Netscape introdujo las cookies en su primera versión de navegador. Desde entonces, el World Wide Web Consortium (W3C) ha sumado las cookies al estándar. Muchos navegadores ahora pueden utilizar las cookies.

Las Cookies son pequeñas piezas de información que el servidor solicita que el navegador las registre en el equipo del cliente. De acuerdo a la especificación original de Netscape una cookie no puede contener más de 4 Kb (el tamaño se refiere a la combinación del nombre y valor de la cookie).

Otras son persistentes, a diferencia de las cookies de sesión, estas pueden sobrevivir meses o aún años. Los navegadores que soportan cookies mantienen uno o más archivos con la información de las cookies.

Como trabaja una Cookie.

Las Cookies son pasadas entre el navegador y el servidor a través de la cabecera HTTP.

El servidor primero crea la cookie usando la cabecera Set-Cookie en una respuesta: subsecuentemente, cada requerimiento del navegador retorna la cookie en la cabecera de la petición. Supongamos que queremos crear una cookie llamada color que contenga como valor la cadena rojo, el Servidor debería enviar una cabecera parecida a esto:

```
Set-Cookie: color=rojo; path=/; domain=issd.edu.ar;  
expires=Tuesday, 01-Jan-11 00:00:01 GMT
```

El navegador agrega una cookie llamada color con el valor rojo. Además la información de la cookie deberá ser enviada al servidor cada vez que el navegador haga una petición de página.

El atributo domain define donde la cookie podrá enviarse. En este ejemplo la cookie se enviará al servidor cuando solicitemos una página en la dirección issd.edu.ar. La cookie no se enviará si hacemos una petición por ejemplo a la dirección google.com.ar o cualquier otro sitio de la Web de Internet.

Finalmente, el atributo expires especifica cuando la cookie deberá eliminarse.

Después que el servidor crea una cookie el navegador retorna la cookie en cada requerimiento que hacemos al sitio Web.

cookie persistente : Creación y acceso.

Una cookie persistente tiene una fecha de caducidad de la misma.

Cuidado: No podemos garantizar que la cookie existirá por el tiempo que hemos fijado al crearla, ya que el navegador puede en cualquier momento borrarla o inclusive el usuario del equipo cliente puede manualmente borrar las cookies.

Haremos dos páginas, en una cargaremos el valor de la cookie y en la segunda la imprimiremos.

La interface y código de la primer página es:

Valor de la cookie "edad":

Crear cookie [Label1]

Para el evento Clic del botón “Crear cookie” escribimos el siguiente código:

```
protected void Button1_Click(object sender,
EventArgs e)
{
    HttpCookie cookie1=new
    HttpCookie("edad",this.TextBox1.Text);
    cookie1.Expires = new DateTime(2018, 12, 25);
    this.Response.Cookies.Add(cookie1);
    this.Label1.Text = "Se creó la cookie";
}
```

Creamos una cookie con el nombre “edad” y el valor almacenado en el TextBox1. Fijamos como fecha de caducidad de la cookie el 25 de diciembre de 2018, por último llamamos al método add del objeto Cookies.

La segunda página tiene por objetivo recuperar la cookie en caso que se haya creado previamente, la interface y código de la misma es:

Valor de la cookie "edad":Label

|

```
protected void Page_Load(object sender, EventArgs e)
{
    if (this.Request.Cookies["edad"]==null)
    {
        this.Label1.Text = "No existe la cookie
edad";
    }
    else
    {
        this.Label1.Text=this.Request.Cookies["edad"].Value;
    }
}
```

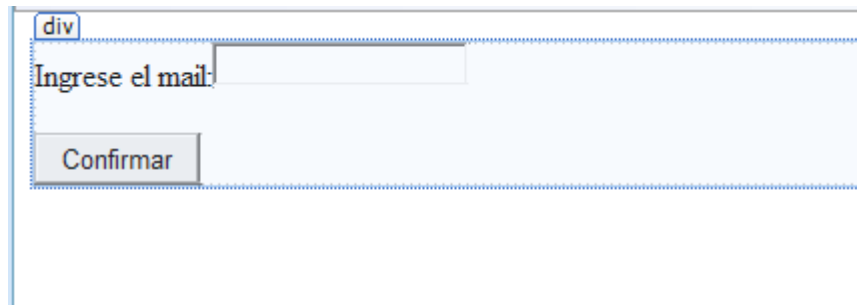
En el evento Load de la página verificamos si existe la cookie, en caso de existir la recuperamos de la propiedad Cookies del objeto Request.

1 – Mostrar el último mail ingresado en un control TextBox.

El objeto del siguiente ejemplo es la creación de una página que solicite el ingreso del mail de una persona, si en otro momento ya lo había ingresado mostrarlo precargado en el control TextBox.

Emplearemos una cookie persistente para almacenar el mail ingresado.

La página y el código es el siguiente:

The image shows a web browser window with a light blue background. Inside the window, there is a text input field with the placeholder text "Ingrese el mail:". Below the input field is a button labeled "Confirmar". The browser's address bar shows a "div" tab.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class Default3 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (this.IsPostBack==false)
        {
            if (this.Request.Cookies["mail"]!=null)
            {
                this.TextBox1.Text=this.Request.Cookies["mail"].Value;
            }
        }
        protected void Button1_Click(object sender,
        EventArgs e)
        {
            HttpCookie cookiel=new
            HttpCookie("mail",this.TextBox1.Text);
            cookiel.Expires=new DateTime(2018,12,25);
            this.Response.Cookies.Add(cookiel);
        }
    }
}
```

En el método Page_Load de la página verificamos si es la primera vez que pedimos la página (es decir que no se recarga por presionar el botón) y si existe la cookie, en dicho caso cargamos el contenido de la propiedad Text del control TextBox.

Luego de ingresar un mail en el control TextBox y presionar el botón “Confirmar” se crea una cookie llamada “mail” y almacena el valor ingresado en el control TextBox. Cerremos el navegador e ingresemos nuevamente a la página, veremos que el mail aparece cargado. (Aunque apaguemos y prendamos la máquina el último mail aparecerá dentro del control)

Como podemos ver una cookie es muy útil si queremos almacenar datos de configuración de un sitio para cada visitante.

cookie de sesión : Creación y acceso.

La diferencia de una cookie de sesión con una persistente es que las cookies de sesión permanecen mientras no cerramos la instancia del navegador, luego el código para crear una cookie de sesión **es similar a las cookies persistentes con la salvedad que no debemos especificar fecha de expiración:**

```
HttpCookie cookie1=new
HttpCookie( "edad",this.TextBox1.Text );
this.Response.Cookies.Add(cookie1);
this.Label1.Text = "Se creó la cookie";
```

Variables de Sesión

Una variable de sesión permite preservar el valor de una variable a través de una serie de páginas. Una variable de sesión se utiliza normalmente para almacenar una preferencia del usuario, un carrito de compras, información de seguridad del usuario, nombres de usuarios, password, etc.

Las variables de sesión son almacenadas durante el tiempo que el usuario visita el sitio Web. Cuando el servidor detecta que el usuario no hace más peticiones de páginas, la información almacenada de las variables de sesión es automáticamente destruida **(por defecto está configurado para que la destrucción de dichas variables suceda luego de 20 minutos de inactividad).**

Podemos modificar el tiempo de vida de las variables de sesión inicializando la propiedad Timeout del objeto Session que tiene toda página (el valor que se asigna representa minutos).

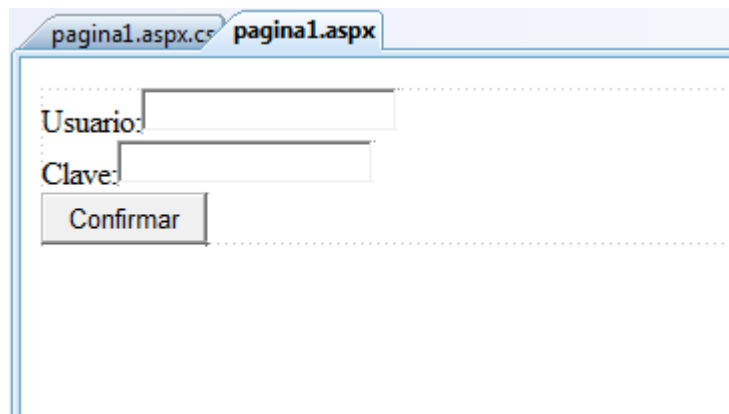
```
this.Session.Timeout = 10;
```

Los datos que podemos almacenar en variables de sesión pueden ser de cualquier tipo: string, integer, ArrayLists, DataSet, DataTable, etc.

Para ilustrar el uso de variables de sesión haremos una serie de páginas donde en la primera ingresaremos el nombre de usuario y clave, en la segunda los listaremos, y en esta dispondremos un hipervínculo para dirigirnos a una tercera página, donde mostraremos nuevamente el contenido de las variables de sesión.

Página 1:

Esta página solicita la carga de los dos datos y redirecciona a la segunda página, la interface visual es la siguiente:

The image shows a web browser window with a single tab titled 'pagina1.aspx'. The page content is a simple login form. It has two text input fields: the first is labeled 'Usuario:' and the second is labeled 'Clave:'. Below these fields is a button labeled 'Confirmar'. The form is enclosed in a light blue border.

El código para el evento Click del botón confirmar es el siguiente:

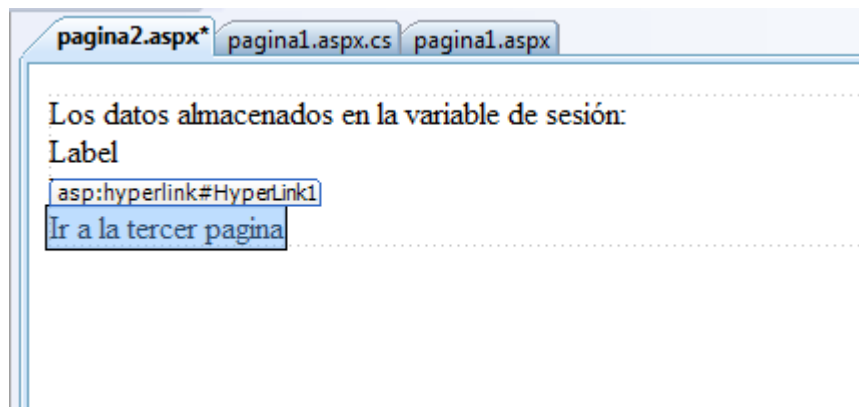
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class paginal : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender,
    EventArgs e)
    {
        //Almacenamos las dos variables de sesion
        this.Session["usuario"] = this.TextBox1.Text;
        this.Session["clave"] = this.TextBox2.Text;
        //Redireccionamos a la segunda pagina
        this.Response.Redirect("pagina2.aspx");
    }
}
```

Página 2:

Tiene por objetivo mostrar los contenidos de las dos variables de sesión. Además hay un hipervínculo (objeto de la clase HyperLink) que llama a la tercera página.

La interface visual es:



El código fuente de esta página es la siguiente:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

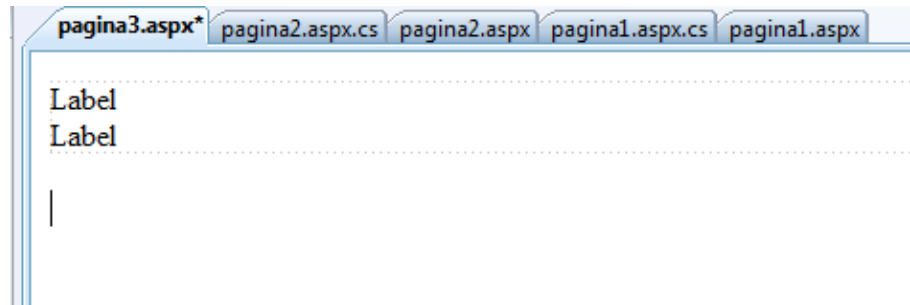
public partial class pagina1 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void Button1_Click(object sender,
EventArgs e)
    {
        //Almacenamos las dos variables de sesion
        this.Session["usuario"] = this.TextBox1.Text;
        this.Session["clave"] = this.TextBox2.Text;
        //Redireccionamos a la segunda pagina
        this.Response.Redirect("pagina2.aspx");
    }
}
```

Es decir inicializamos las Label con los contenidos de las variables de sesión que están almacenadas en memoria y administradas por el servidor Web.

Página 3:

Por último esta tercer página tiene por objetivo mostrar nuevamente el contenido de las variables de sesión.

La interface es:



Y el código fuente del evento Load de la página es:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

public partial class pagina3 : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        //Introducir aquí el código de usuario
        //y la clave
        Label1.Text =
this.Session["usuario"].ToString();
        Label2.Text = this.Session["clave"].ToString();
    }
}
```

Las variables de sesión se verifican su existencia igual que las cookies.

Ejercicios Propuestos

1 – Confeccionar un sitio que muestre noticias Deportivas, Políticas y Culturales. Mediante una página de configuración permitir que un visitante pueda especificar que tipo de noticias quiere ver en la portada del sitio (emplear tres cookies)