

Departament de Llenguatges i Sistemes Informàtics

Tema 10. Capa d'Interfície II

Herramientas Avanzadas para el Desarrollo de Aplicaciones

Escola Politècnica Superior
Universitat d'Alacant

Indice

1. Controls de llista senzills
2. Controls de Navegació
3. Controls de Validació
4. Objectes Session i Application
5. Esdeveniments d'Aplicació: Global.asax
6. Distribució de capes en el projecte

1

Controls de
llista senzills

Controls de Llista Senzills

DROPDOWNLIST

rojo ▼

LISTBOX

rojo
azul
verde

CHECKBOXLIST

☐ rojo
☐ azul
☐ verde

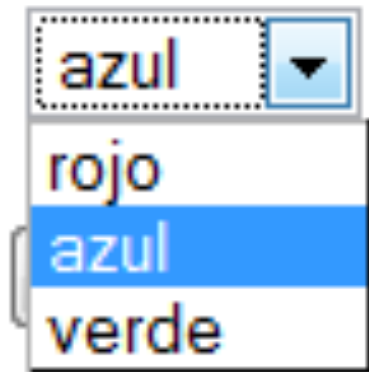
RADIOBUTTON LIST

☐ rojo
☐ azul
☐ verde

- Llista desplegable
- Llista desplegada
- Llista de botons de verificació
- Llista de botons de ràdio

Controls de Llista Senzills (listBox, dropDownList, radioButtonList, checkBoxList)

- Llista desplegable (dropDownList)



- ElementoLista1 (ListItem) FILA 0
- ElementoLista2 *Seleccionado* (ListItem) FILA 1
- ElementoLista3 (ListItem) FILA 2

Afegir elements de forma declarativa

```
<asp:DropDownList ID="DropDownList1" runat="server"
onselectedindexchanged="DropDownList1_SelectedIn
dexChanged" Width="90px" AutoPostBack="False">
    <asp:ListItem Value="rojo1">rojo</
asp:ListItem>
    <asp:ListItem>azul</asp:ListItem>
    <asp:ListItem>verde</asp:ListItem>
</asp:DropDownList>
```

RepeatDirection

- RadioButtonList: Ilista de botons de radio

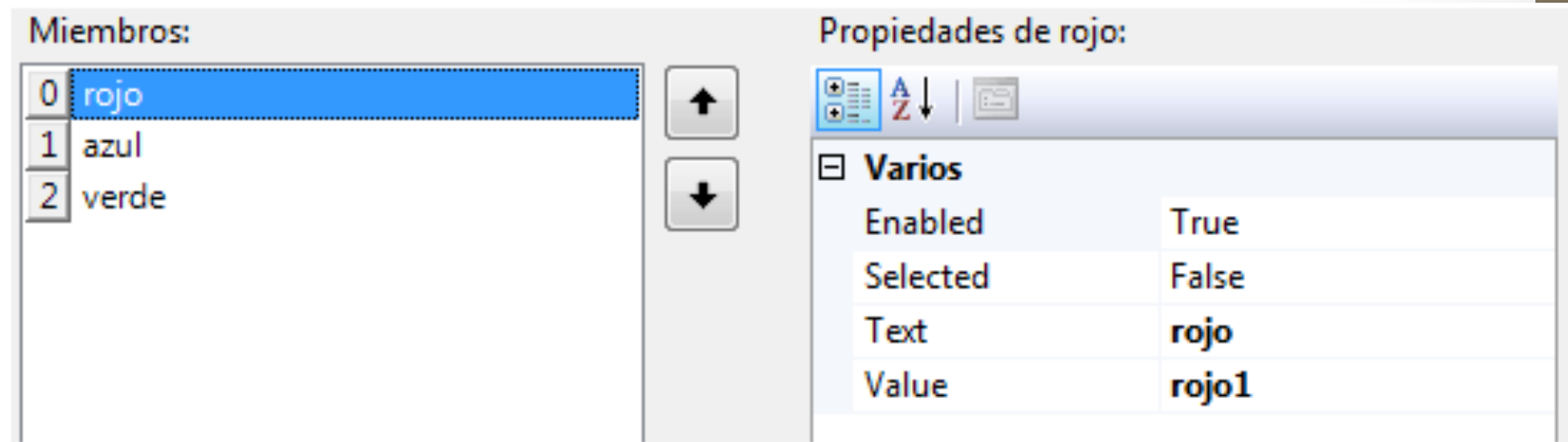
```
<asp:RadioButtonList ID="RadioButtonList1" runat="server"
    onselectedindexchanged="RadioButtonList1_SelectedIndexChanged">
    <asp:ListItem>rojo</asp:ListItem>
    <asp:ListItem>azul</asp:ListItem>
    <asp:ListItem>verde</asp:ListItem>
</asp:RadioButtonList>
```

- CheckBoxList: Ilista de opciones múltiple

```
<asp:CheckBoxList ID="CheckBoxList1" runat="server"
    onselectedindexchanged="CheckBoxList1_SelectedIndexChanged"
    RepeatDirection="Horizontal">
    <asp:ListItem>rojo</asp:ListItem>
    <asp:ListItem>azul</asp:ListItem>
    <asp:ListItem>verde</asp:ListItem>
</asp:CheckBoxList>
```

Propietat Items: col·lecció de ListItem

Objecte ListItem: propietats



- Text:
 - contingut visualitzat
- Value:
 - valor ocult del codi HTML
- Selected:
 - true o false (seleccionat o no)

Propietats dels Controls de Llista Senzills

- Propietat **SelectedIndex**
 - Indica la fila seleccionada com un índex que comença en zero
- Propietat **SelectedItem**
 - Permet que el codi recuperi un objecte **ListItem** que representa l'element seleccionat

```
Label1.Text = DropDownList1.SelectedIndex.ToString();
```

```
Label1.Text = DropDownList1.SelectedItem.Text.ToString();
```

```
Label1.Text = DropDownList1.SelectedItem.Value.ToString();
```

```
Label1.Text=DropDownList1.SelectedValue;
```

Controls de llista amb selecció múltiple

- **ListBox:**
 - Poden seleccionar-se diversos elements: propietat **SelectionMode=Multiple**
- **CheckBoxList**
 - Sempre poden seleccionar-se diversos elements
- Per trobar tots els elements seleccionats necessitem
 - recórrer la col·lecció Items del control llesta
 - comprovar la propietat ListItem.Selected de cada element

```
Foreach ListItem i in DropDownList1.Items  
    { if (i.Selected==true)  
        ....    }
```

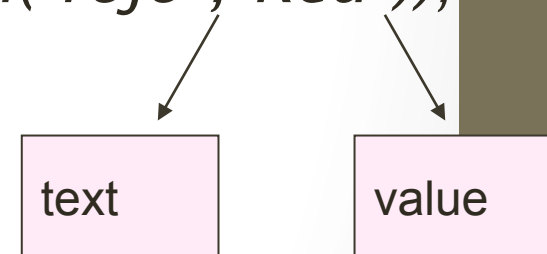
Afegir elements dinàmicament a una llista (codi)

- Mètode *Add* de l'objecte *Items*

EJEMPLO:

DropDownList1.Items.Add("rojo");

DropDownList1.Items.Add(new ListItem("rojo", "Red"));

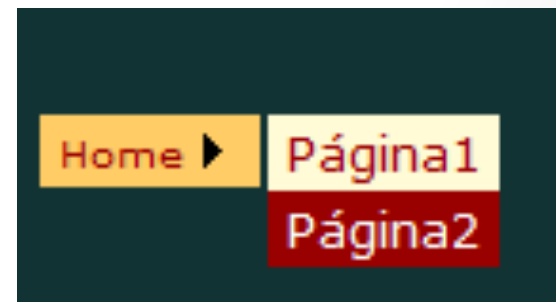


2

Controls de
navegació:
Menu

Control meno

- Es pot utilitzar per crear un menú que col·locarem a la pàgina mestra i altres ajudes de navegació.
- Permet afegir un menú principal amb submenús i també ens permet definir menús dinàmics.
- Els elements del Menu poden afegir-se directament en el control o enllaçar-los amb una font de dades.
- En les propietats podem especificar l'aparença, orientació i contingut del menú.



Static Display y Dynamic Display

- El control té dues maneres de “**Display**”:
 - **Estàtic (static)**: El control Menu està expandit completament tot el temps. Tota l'estructura és visible i l'usuari pot fer clic en qualsevol part.
 - **Dinàmic (dynamic)**: En aquest cas solament són estàtiques les porcions especificades, mentre que els elements fills es mostren quan l'usuari manté el punter del ratolí sobre el node pare.

Propietats

- **StaticDisplayLevels.**
 - Aquesta propietat permet especificar el nombre de nivells estàtics que volem mostrar com a arrel del menú (el mínim és 1).
- **MaximumDynamicDisplayLevels**
 - Aquesta propietat especifica quants nivells de nodes que apareixen de forma dinàmica es mostressin després del nivell estàtic.

Dynamic: cantidad de tiempo

- També podem especificar la quantitat de temps que volem que tarda la part dinàmica d'un menú a desaparèixer.
- El podem especificar en mil·lisegons mitjançant la propietat **DisappearAfter** del menu:
 - `Menu.DisappearAfter=1000;`
- El valor per defecte són 500 ms.

Definició del contingut de menu

- Afegir objectes individuals MenuItem (de forma declarativa o programàtica) . També se li pot enllaçar un arxiu XML.
- Propietats del control → propietat Items (col·lecció d'objectes MenuItem)

```
<asp:Menu ID="Menu1" runat="server" StaticDisplayLevels="3">
```

```
<Items>
```

```
<asp:MenuItem Text="File" Value="File">
```

```
<asp:MenuItem Text="New" Value="New"></asp:MenuItem>
```

```
<asp:MenuItem Text="Open" Value="Open"></asp:MenuItem>
```

```
</asp:MenuItem>
```

```
<asp:MenuItem Text="Edit" Value="Edit">
```

```
<asp:MenuItem Text="Copy" Value="Copy"></asp:MenuItem>
```

```
<asp:MenuItem Text="Paste" Value="Paste"></asp:MenuItem>
```

```
</asp:MenuItem>
```

```
<asp:MenuItem Text="View" Value="View">
```

```
<asp:MenuItem Text="Normal" Value="Normal"></asp:MenuItem>
```

```
<asp:MenuItem Text="Preview" Value="Preview"></asp:MenuItem>
```

```
</asp:MenuItem>
```

```
</Items>
```

```
</asp:Menu>
```

Code-Behind C#

```
protected void Menu1_MenuItemClick(object sender,
    System.Web.UI.WebControls.MenuEventArgs e)
{
    switch(e.Item.Value)
    {
        case "Products":
            ...
            return;
        case "Services":
            ...
            return;
    }
}
```

[http://msdn.microsoft.com/en-us/library/16yk5dbv\(v=vs.80\).aspx](http://msdn.microsoft.com/en-us/library/16yk5dbv(v=vs.80).aspx)

http://www.obout.com/em/doc_server.aspx

3

Controls de Validació

Validació de dades

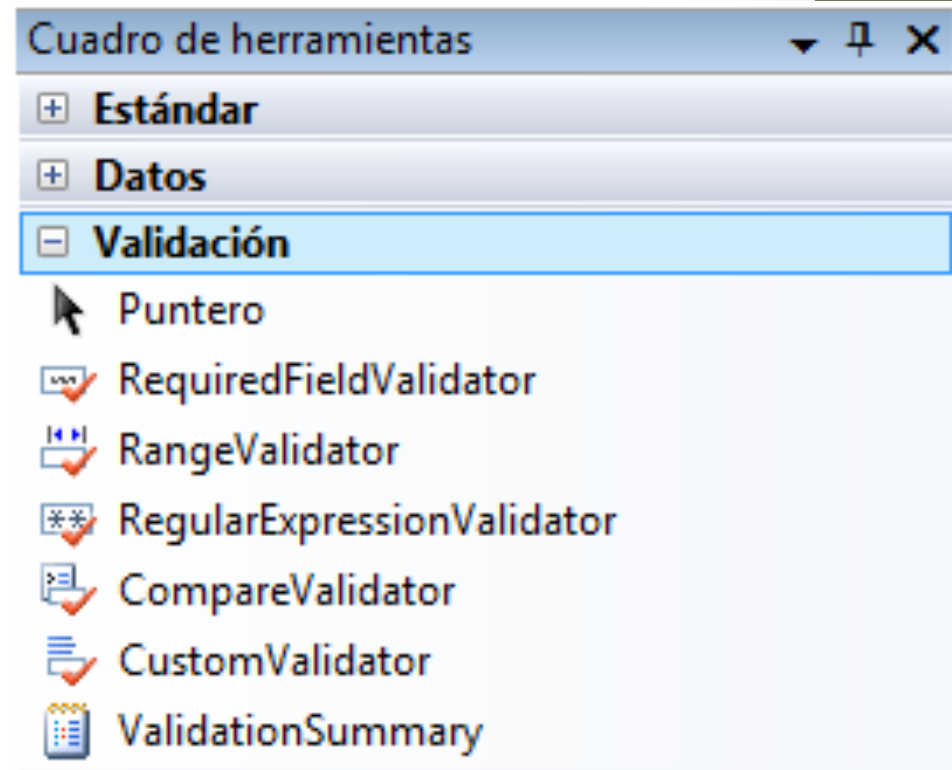
- Hem d'assegurar que els usuaris introdueixen dades correctament
 - Adreça d'email
 - Número de telèfon
- Asp.net proporciona un conjunt de controls de validació predefinits
- Dos tipus de validació
 - Validació del costat del client
 - Validació del costat del servidor

Validació de dades

- Costat del client:
 - Utilització de codi Javascript que valida les dades introduïdes per l'usuari, directament en el navegador
- Costat del servidor:
 - Utilització de codi (C#) per validar les dades de formularis una vegada han estat enviats al servidor

Controls de validació

- Asp.net detecta si el navegador suporta validació del costat del client:
 - Generen el codi Javascript necessari per validar les dades
 - En un altre cas, les dades del formulari es validen en el servidor



Controls de validació

- Sobre diferents controlis web:
 - TextBox
 - ListBox
 - DropDownList
 - RadioButtonList
 - FileUpload

Controls de Validació

- Propietats
 - Per mostrar el missatge d'error **ErrorMessage**
 - Per indicar el control a validar **ControlToValidate**
- **Entrada requerida:** RequiredFieldValidator: La validació és OK quan el control d'entrada no conté una cadena buida.
 - InitialValue
- **Coincidència de models:** RegularExpressionValidator: La validació és OK si el valor d'un control d'entrada es correspon amb una expressió regular especificada.
 - ValidationExpression

Controls de validació

```
<form id="form1" runat="server">
  <div>
    Usuari: <asp:TextBox ANEU="TextBox1" runat="server">
      </asp:TextBox>
    <asp:RequiredFieldValidator ID="UserNameReq" runat="server"
      ControlToValidate="TextBox1"
      ErrorMessage="Introduce el usuario!!"> </asp:RequiredFieldValidator>
    Password: <asp:TextBox ID="TextBox2" runat="server">
      </asp:TextBox>
    <asp:RequiredFieldValidator ID="PasswordReq" runat="server"
      ControlToValidate="TextBox2"
      ErrorMessage="Introduce el password!!"> </asp:RequiredFieldValidator>
    <asp:Button ID="Button1" runat="server" Text="Enviar" />
  </div>
</form>
```

Usuario: Introduce el usuario!!

Password: Introduce el password!!

Enviar

Controls de Validació

- **Comparación amb un valor:** CompareValidator: La validació és OK si el control conté un valor que es correspon amb el valor d'un altre control especificat.
 - ControlToCompare
 - ControlToValidate
 - ValueToCompare
 - Type
 - Operator
- **Comprobación d'interval:** RangeValidator: La validació és OK quan el control d'entrada conté un valor dins d'un interval numèric, alfabètic o temporal especificat.
 - MaximumValue
 - MinimumValue
 - Type

Controls de Validació

- CustomValidator: La validació la realitza una funció definida per l'usuari.
 - ClientValidationFunction
 - OnServerValidate
- **Suposem que volem validar que el nom d'usuari sigui únic en l'aplicació**

```
<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>  
<asp:CustomValidator ID="CustomValidator1"  
    ControlToValidate="TextBox1"  
    OnServerValidate="ComprobarUsuario" runat="server"  
    ErrorMessage="El usuario ya existe!!">  
</asp:CustomValidator>
```

Controls de validació

```
protected void Button1_Click(object sender, EventArgs e)
{
    if (Page.IsValid)
    {   Button1.Text = "Correcto"; }
    else
    {   Button1.Text = "Incorrecto"; }
}
```

```
protected void ComprobarUsuario(object sender,
    ServerValidateEventArgs e)
{
    string username = e.Value.ToLower();
    if (username == "sonia" || username == "irene")
    { e.IsValid = false; }
}
```

Controls de Validació

- ValidationSummary: Aquest control mostra un resum amb tots els missatges d'error de cada control de validació.

- ShowMessageBox: mostra un quadre de diàleg resumeixen
- ShowSummary

The image shows a web form with three input fields: 'Usuario:', 'Password:', and 'Confirma:'. The 'Usuario:' and 'Password:' fields have red error messages next to them: 'Introduce el usuario!!' and 'Introduce el password!!' respectively. The 'Confirma:' field has a red dot next to it. A dialog box titled 'La página en http://localhost:49999 dice:' is open, showing a yellow warning icon and the same two error messages. The dialog box has an 'Aceptar' button. At the bottom of the form, there is an 'Enviar' button.

Usuario: Introduce el usuario!!

Password: Introduce el password!!

Confirma:

• Introduce el usuario!!

• Introduce el password!!

Aceptar

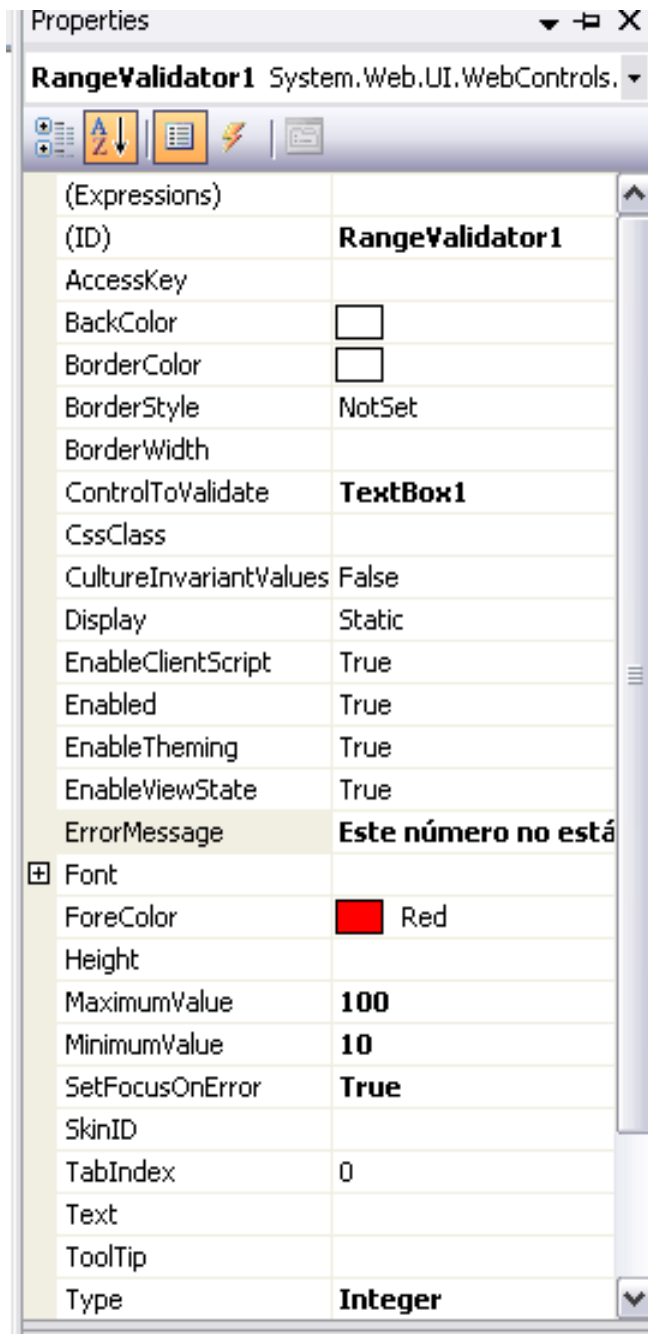
Enviar

Propiedad Display

- Serveix per comprovar si el control de validació mostra missatges d'error. La propietat Display pot establir-se en :
- **None** — El control de validació no apareix a la pàgina.
- **Static** — Cada control de validació ocupa espai encara que no sigui visible el text d'un missatge d'error, la qual cosa permet definir una presentació fixa per a la pàgina.
- **Dynamic** — Els controls de validació no ocupen espai tret que mostrin un missatge d'error, la qual cosa els permet compartir la mateixa ubicació. No obstant això, quan es mostra el missatge d'error, la presentació de la pàgina canvia, la qual cosa de vegades produeix que els controls canviïn les posicions.
- La presentació dinàmica requereix un explorador que admeti Html dinàmic (DHTML).

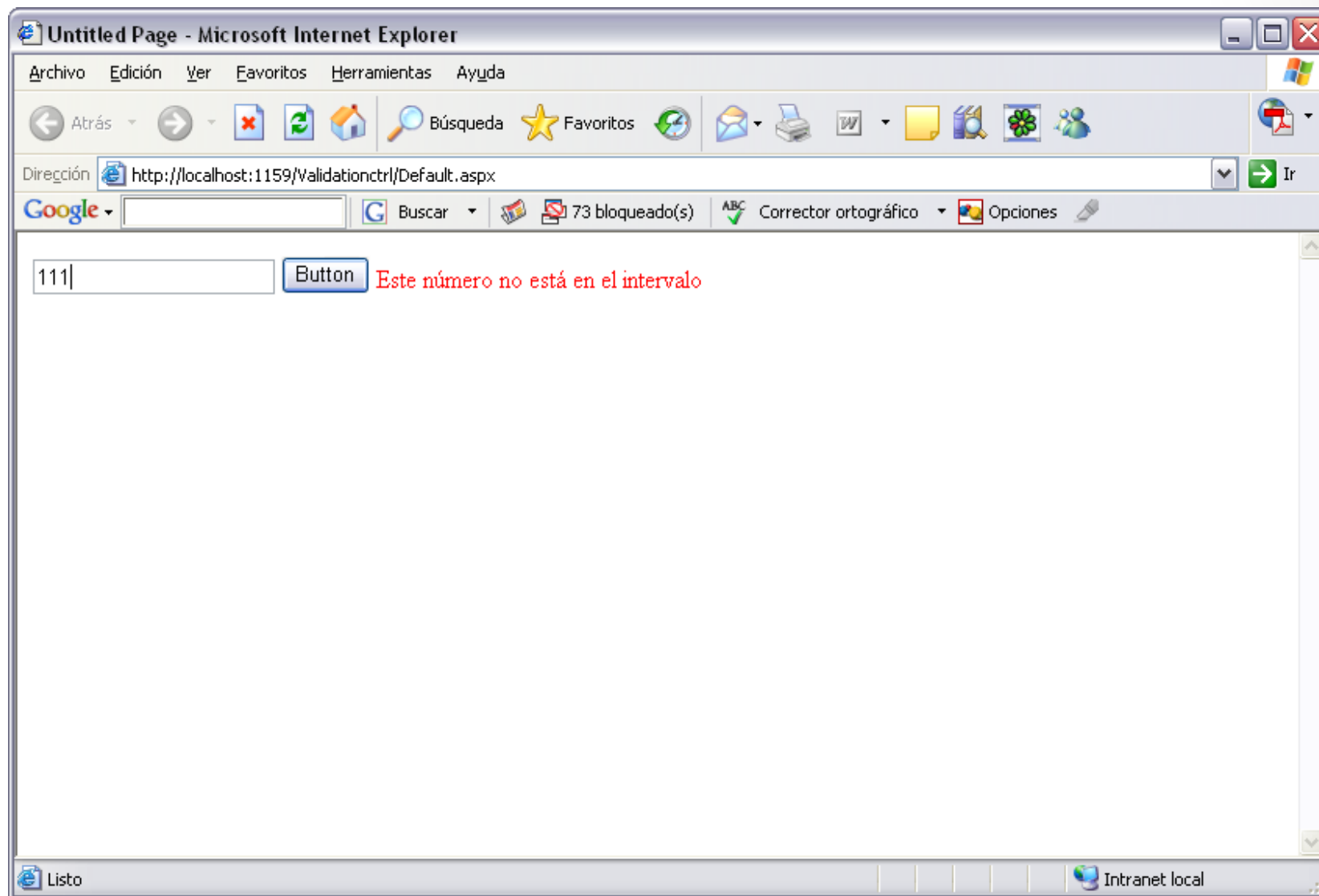
El procés de validació (servidor)

1. L'usuari rep una pàgina i comença a emplenar els valors d'entrada.. Al final l'usuari prem un botó per enviar la pàgina.
2. Cada control Button té una propietat *Causis Validation*.
 - Si aquesta propietat és **False**, ASP.NET ignora els controls de validació.
 - Si està a **True** (Valor Predeterminat), ASP.NET valida automàticament la pàgina quan l'usuari prem el botó. Es realitza la validació de cada control de la pàgina.
 - Cada control de validació exposa la seva **propietat IsValid**, La pàgina també exposa una propietat IsValid que resumeix l'estat **IsValid** de tots els controls de validació de la pàgina.



```
<asp:RangeValidator  
  ID="RangeValidator1 "  
  runat="server"  
  ControlToValidate="TextBox1 "  
  ErrorMessage="Aquest nombre  
no està en l'interval"  
  MaximumValue="100"  
  MinimumValue="10"  
  Type="Integer">  
</asp:RangeValidator>
```


Vesteixi execució (range validator)



Expressions regulars

- **Adreça de correu electrònic**
 - Comprovar que existeix una @, un punt i només permet caràcters que no siguin espais.
- **Contrasenya**
 - Entre 4 i 10 caràcters i el primer caràcter ha de ser una lletra.
- **Número de compte**
 - Seqüència de 4, 4, 2, i 10 dígit, cada grup separat per un guió.
- **Camp de longitud limitada**
 - Entre 4 i 10 caràcters incloent caràcters especials (*, &...)

Sintaxis Expresiones Regulares

- zero o més ocurrències del caràcter o subexpresión anterior.
- + una o més ocurrències del caràcter o subexpresión anterior
- () agrupa una subexpresión que es tracta com un únic element
- | Qualsevol de les dues parts (OR)
- [] es correspon amb un caràcter en un interval de caràcters vàlids [a-c]
- {n} exactament n dels caràcters o subexpresiones anteriors
- . qualsevol caràcter excepte el salt de línia
- ? el caràcter anterior o la subexpresión anterior és opcional
- ^ començament d'una cadena
- \$ fi d'una cadena

■ ■ ■

- \s caràcter d'espai en blanc (ex. tab o espai)
- \S qualsevol caràcter no espai
- \d qualsevol caràcter numèric
- \D qualsevol caràcter no dígit
- \w qualsevol caràcter alfanumèric (lletra, nombre o caràcter de subratllat)

Solució...

- Correu electrònic
 - `\S+@\S+\.\S+`
- Contrasenya
 - `[a-zA-Z]\w{3,9}`
- Número de compte
 - `\d{4}-\d{4}-\d{2}-\d{10}`
- Camp de longitud limitada
 - `\S{4,10}`

Grups de validació

- Els controls de validació es poden associar en grups de validació a fi que els controls que pertanyin a un grup comú es validin junts.
- Pot utilitzar aquests grups per habilitar o deshabilitar de forma selectiva la validació per a controls relacionats en una pàgina.
- Cal definir el nom del grup en els controls de validació i en el botó o altres controls d'enviament que causen validació.

SetFocusOnError

- S'estableix en controls de validació que causen que el primer control no vàlid rebi el focus.
- Propietat dels controls de validació.

- Més informació sobre aquests controls

[http://msdn.microsoft.com/es-es/library/debza5t0\(v=vs.90\).aspx](http://msdn.microsoft.com/es-es/library/debza5t0(v=vs.90).aspx)

http://www.elguille.info/colabora/NET2005/FernandoLuque_ASPValidar.htm

4

Manteniment
d'estat :
Objetos Session i
Application

Objectes Session i Application

- Els objectes Session estan associats a un usuari particular i serveixen com a manera de transportar i mantenir les dades de l'usuari en pàgines web, com a fòrums o llocs de comerç electrònic.
- Els objectes Application són compartits per tots els usuaris i permeten emmagatzemar informació compartida per tota l'aplicació web.
- En ASP.NET els objectes Session i Application estan implementats com a col·leccions o conjunts de parells nomeni-valor.

Què és una sessió?

- Una **sessió** és el període de temps en el qual un usuari particular interactua amb una aplicació web.
- Durant una sessió la identitat única d'un usuari es manté internament.
- Les dades s'emmagatzemen temporalment en el servidor.
- Una sessió finalitza si hi ha un *timeout* o si tu finalitzes la sessió del visitant en el codi.

Quin és l'ús d'una sessió?

- Les sessions ajuden a preservar les dades entre accessos successius. Això pot fer-se gràcies als objectes de sessió.
- Els objectes de Sessió ens permeten preservar les preferències de l'usuari i una altra informació de l'usuari en navegar per l'aplicació web.
- Exemple
- Website de comerç electrònic on el visitant navega a través de moltes pàgines i vol seguir què productes ha adquirit.

Objecte Session

- **Session**: serveix per emmagatzemar dades pertanyents a un únic usuari (en l'àmbit d'una sessió).

```
//Esborra tots els valors d'estat de la sesion  
Session.Clear();  
Session.Add("nombre","Homer");  
Response.Write(Session["nombre"]);
```

En ASP.NET

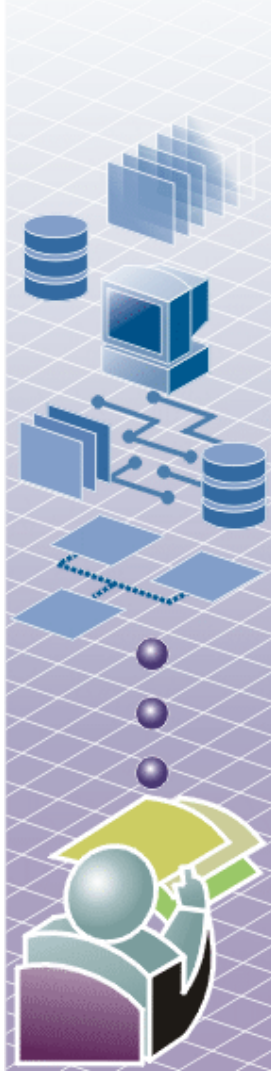
- Les sessions són taules Hash en memòria amb un timeout especificat.
- `Session["username"] = "Jose Martínez";`
`Session["color"] = "Blue";`
- Assignem els valors a les variables de sessió "username" i "color", respectivament. Si necessito saber el "username" o "color" en pàgines següents puc usar `Session["username"]`, `Session["color"]`.
- Les sessions en ASP.NET estan identificades usant sencers 32-bit long coneguts com Session IDs. El motor ASP genera aquests session ANEU's de tal forma que es garanteixi que són únics

Objecte Session

Session Type	Què fa	Exemple
Session.Abandon	Abandona (cancel·la) la sessió actual	
Session.Remove	Esborra un element de la col·lecció de estat de la sessió.	Session["username"] = "Jose Martínez"; (Inicializa una variable de sesión) Session.Remove["username"]; (Borra la variable de sesión "username")
Session.RemoveAll	Esborra tots els elements de estat de la sessió.	

Session Type	Què fa	Exemple
Session.Timeout	Estableix el the timeout (<i>en minuts</i>) para una sessió	Session.Timeout=30 (Si un usuari NO demana una pàgina en l'aplicació ASP.NET en 30 minuts la sessió expira.)
Session.SessionID	Recupera l'ID de la sessió (propietat de només lectura d'una sessió) per a la sessió.	
Session.IsNewSession	És per comprovar que la sessió de l'usuari es va crear amb la petició actual p. ex. l'usuari acaba d'entrar al lloc web. La propietat IsNewSession és certa a la primera pàgina de l'aplicació.	

Exercici



Exercici

- ▶ Crear una aplicació web en la qual demanem introduir un nom d'usuari i un botó enviar.
- ▶ En punxar en el botó que ens redireccione a un segon formulari en el qual posarem “hola” seguit del login introduït,
 - Ficant el login en una variable de sessió

Botón con SESSION

Arxiu Default.aspx.cs

```
protected void Button1_Click (object sender,  
    EventArgs e)  
{  
    Session["login"] = TextBox1.Text;  
    Response.Redirect("session2.aspx");  
}
```

Archivo session2.aspx.cs

```
protected void Page_Load(object sender, EventArgs e)  
{  
    Label3.Text = Session["Login"].ToString();  
}
```

Important

En crear la part privada de la Web utilitzarem variables de sessió per controlar si l'usuari ha entrat logueandose o ha entrat posant directament la URL, en aquest cas la variable de sessió estarà buida i no hauríem de permetre l'accés

Variables d'aplicació

- I si volem inicialitzar variables que estiguin disponibles en una sessió i siguin les mateixes per a tots els usuaris??
- Això suposa que un canvi en el valor d'una **variable d'aplicació** es reflecteix en les sessions actuals de tots els usuaris.

Objecte Application

- Per exemple:
 - Es pot donar un valor a una variable d'aplicació anomenada SiteName
`Application["SiteName"] = "Mi aplicación";`
 - Qualsevol pàgina de l'aplicació pot llegir aquesta cadena:
`string appName = (string)Application["SiteName"];`
- Per eliminar qualsevol variable de l'objecte Application:
`Application.Remove("SiteName");`
- Per eliminar totes les variables:
`Application.RemoveAll();`

Variables d'aplicació



Exercici

- ▶ Crear una aplicació web que expliqui el nombre de visites que rep
 - ▶ Utilitzar variables d'aplicació
 - ▶ Quan arribi a 10 visites el comptador s'ha de reiniciar

Variables d'aplicació

- Primer pas:
 - A la pàgina Default.aspx incloem una etiqueta
`<asp:Label ID="LabelCont" runat="server"></asp:Label>`
- Segon pas:
 - En l'arxiu Default.aspx.cs (code behind) utilitzar una variable Application per controlar el nombre de visites

```
protected void Page_Load(object sender, EventArgs e) {  
    if (Application["PageCounter"] != null && (int)Application["PageCounter"] >= 10)  
    { Application.Remove("PageCounter"); }  
    if (Application["PageCounter"] == null)  
    { Application["PageCounter"] = 1; }  
    else  
    { Application["PageCounter"] =  
        (int)Application["PageCounter"] + 1; }  
    LabelCont.Text = Application["PageCounter"].ToString();  
}
```

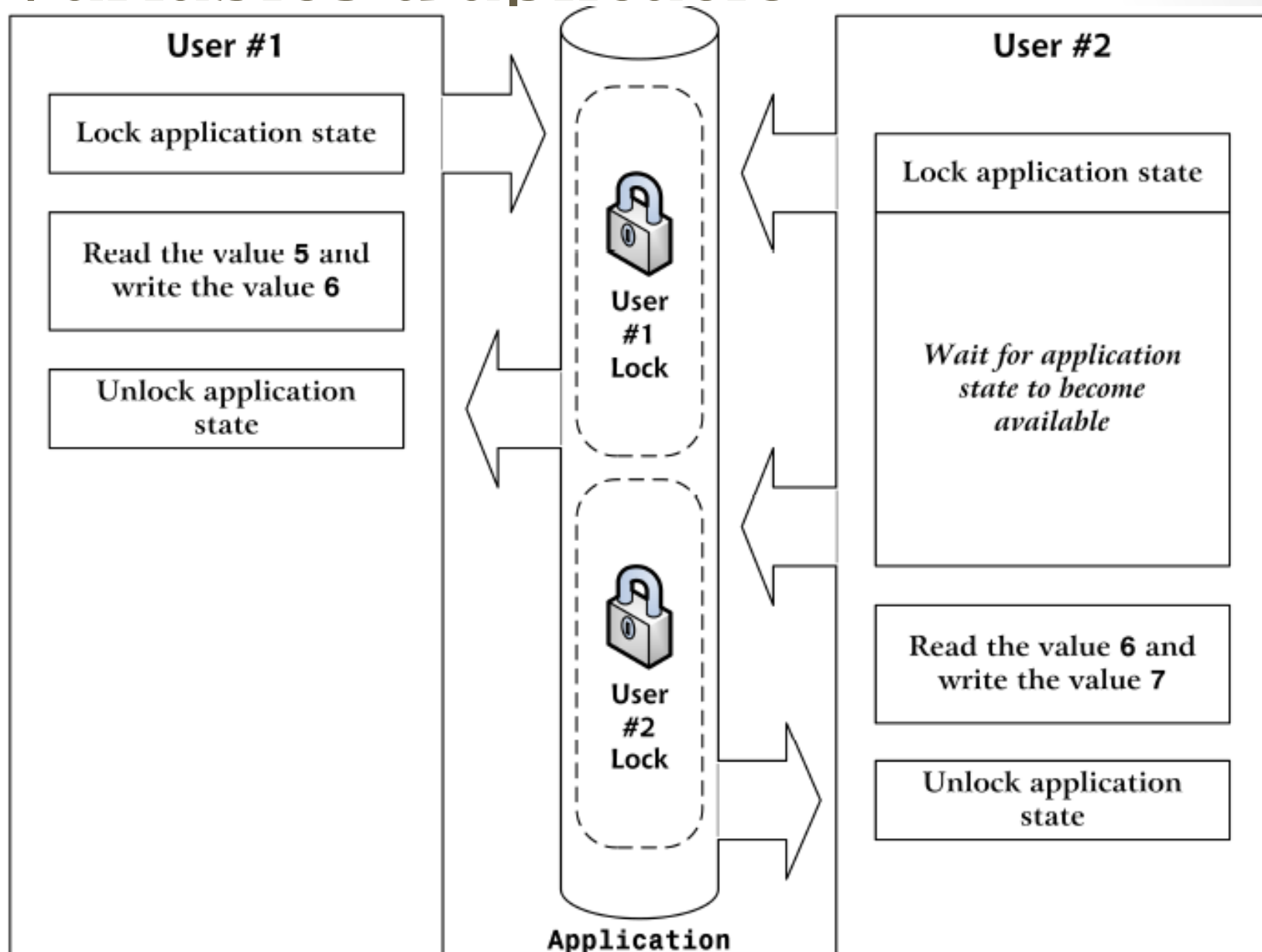
Variables d'aplicació

- Problema:
 - Dues persones carreguen simultàniament la pàgina:
 - El comptador podria incrementar-se només 1 unitat
 $\text{Application["PageCounter"]} = (\text{int})\text{Application["PageCounter"]} + 1$
 - L'expressió de la dreta s'avalua primer
 - L'usuari1 llegeix el valor de PageCounter emmagatzemat en l'aplicació
 - L'usuari2 llegeix el valor de PageCounter emmagatzemat en l'aplicació
 - Tots dos li sumen 1 unitat però l'increment final en lloc de ser 2 unitats és només 1

Actualitzar una variable d'aplicació

- Solució:
 - En un instant concret, diverses sessions poden estar intentant canviar el valor, encara que només una sessió estarà autoritzada a canviar-ho.
 - ASP.NET té exclusió mútua per a aquest tipus de problemes:
 - `Application.Lock()` – Bloqueja les variables d'aplicació
 - `Application.Unlock()` – Desbloqueja les variables d'aplicació
 - Una vegada que les variables estan bloquejades les sessions que intenten canviar-les han d'esperar.

Variables d'aplicació



Objeto Application

- **Application**: proporciona una manera sencilla d'emmagatzemar en el servidor dades comunes a tots els visitants de nostre assetjo web.

```
Application.Lock();
```

```
Application.Add("edad",22);
```

```
int valor=(int)Application["edad"];
```

```
valor++;
```

```
Application["edad"]=valor;
```

```
Application.Unlock();
```

```
Response.Write(Application["edad"]);
```

Problema

- On inicialitzo una variable d'aplicació perquè no es reiniciï cada vegada (ex. Comptador de visites)????

5

Global.asax



L'arxiu global.asax

- Permet escriure codi d'aplicació global.
- No conté etiquetes HTML ni ASP.NET
- S'utilitza per definir variables globals i reaccionar a esdeveniments globals.
- Conté codi de tractament d'esdeveniments que reacciona als esdeveniments d'aplicació o sessió.

L'arxiu global.asax

- S'afegeix a l'aplicació web com un nou element Classe d'aplicació global
 - Global.asax y Global.asax.cs

```
protected void Application_Start(object sender, EventArgs e)
{ Application["SiteName"] = "Mi aplicación"; }
```

```
protected void Session_Start(object sender, EventArgs e)
{
    Session.Timeout = 15;
    Response.Write("Servida el " + DateTime.Now.ToString());
}
```

- Important:
 - Qualsevol canvi en l'arxiu global.asax reiniciarà l'aplicació

Exemple, ús d'objectes de sessió

- Es vol modificar el timeout per defecte (20min)
- Es pot fer en qualsevol lloc del codi però el més recomanable és fer-ho en l'arxiu Global.asax

Archivo Global.asax

```
protected void Session_Start(object sender, EventArgs e)
{
    Session.Timeout = 15;
}
```

L'arxiu global.asax

- Només pot haver-hi un arxiu global.asax
- Ha de residir en el directori arrel de l'aplicació

Global.asax.cs

```
protected void Application_Error(object sender, EventArgs e)
{
    Response.Write("<b>");
    Response.Write("Oops! Ha ocorregut un error! </b>");
    Response.Write(Server.GetLastError().Message.ToString());
    Response.Write(Server.GetLastError().ToString());
    Server.ClearError();
}
```

Oops! Ha ocorregut un error! System.Web.HttpUnhandledException: Se produjo una excepción de tipo 'System.Web.HttpUnhandledException'. --->
System.DivideByZeroException:

Default.aspx.cs

```
int j = 1;
int x = 0;
int k=j / x;
```


COMPTADOR DE VISITES

Afegir nou element...

- Classe d'aplicació Global → Global.aspx

```
void Application_Start(object sender, EventArgs e)
{
```

```
// Codi que s'executa en iniciar-se l'aplicació
```

```
Application.Add("contador", 0);
```

```
}
```

6

Distribució capes
en el projecte

En la pràctica en grup...

- Una solució amb 2 projectes:
 - Projecte Web (amb referència al projecte de llibreria)
 - Contindrà la interfície i validacions
 - Projecte de Llibreria de classes amb les Capes EN i CAD
 - Carpeta EN
 - Classes EN para entitats del sistema
 - Carpeta CAD
 - Per cada EN una classe CAD per a l'accés a dades