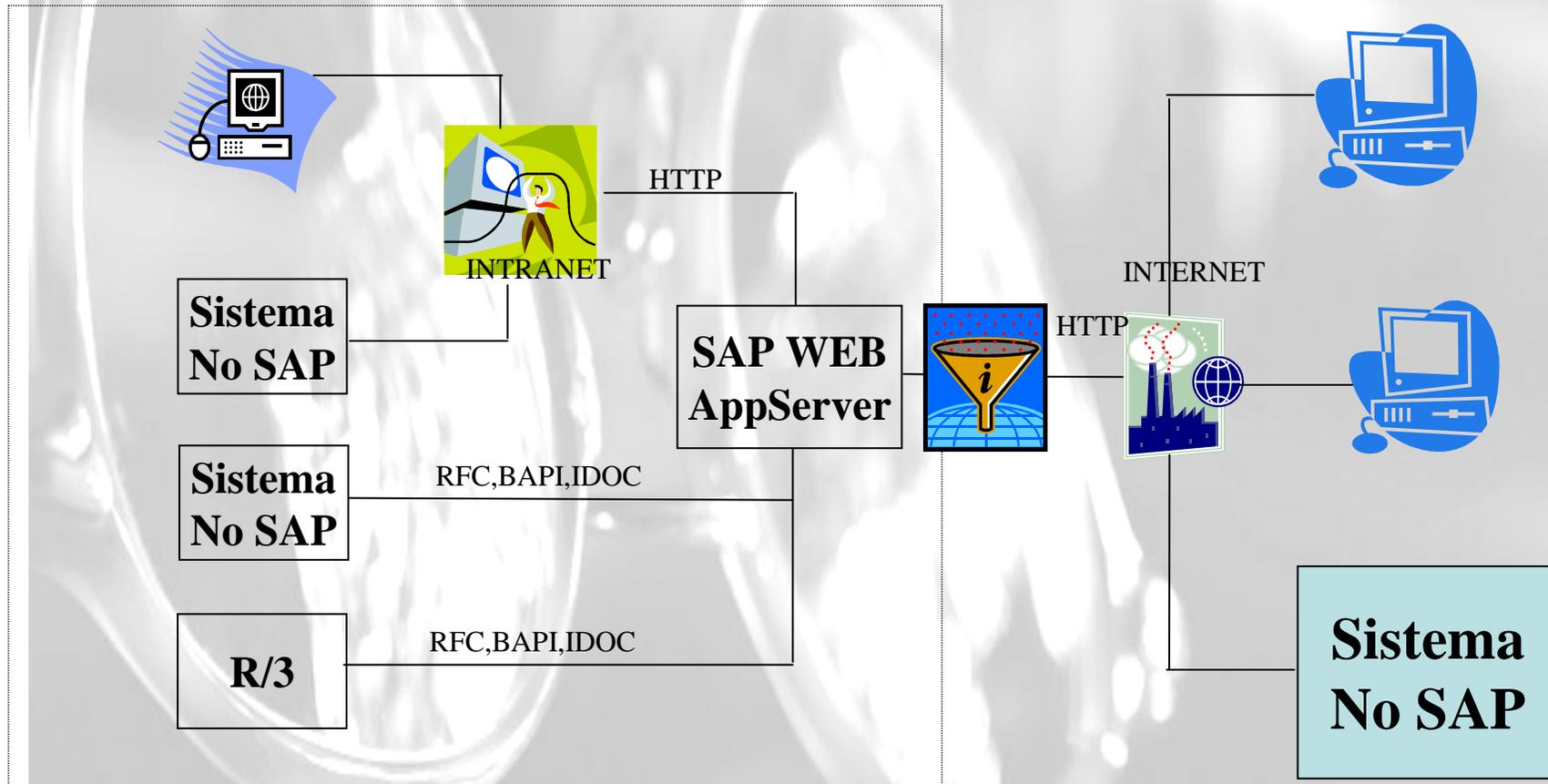
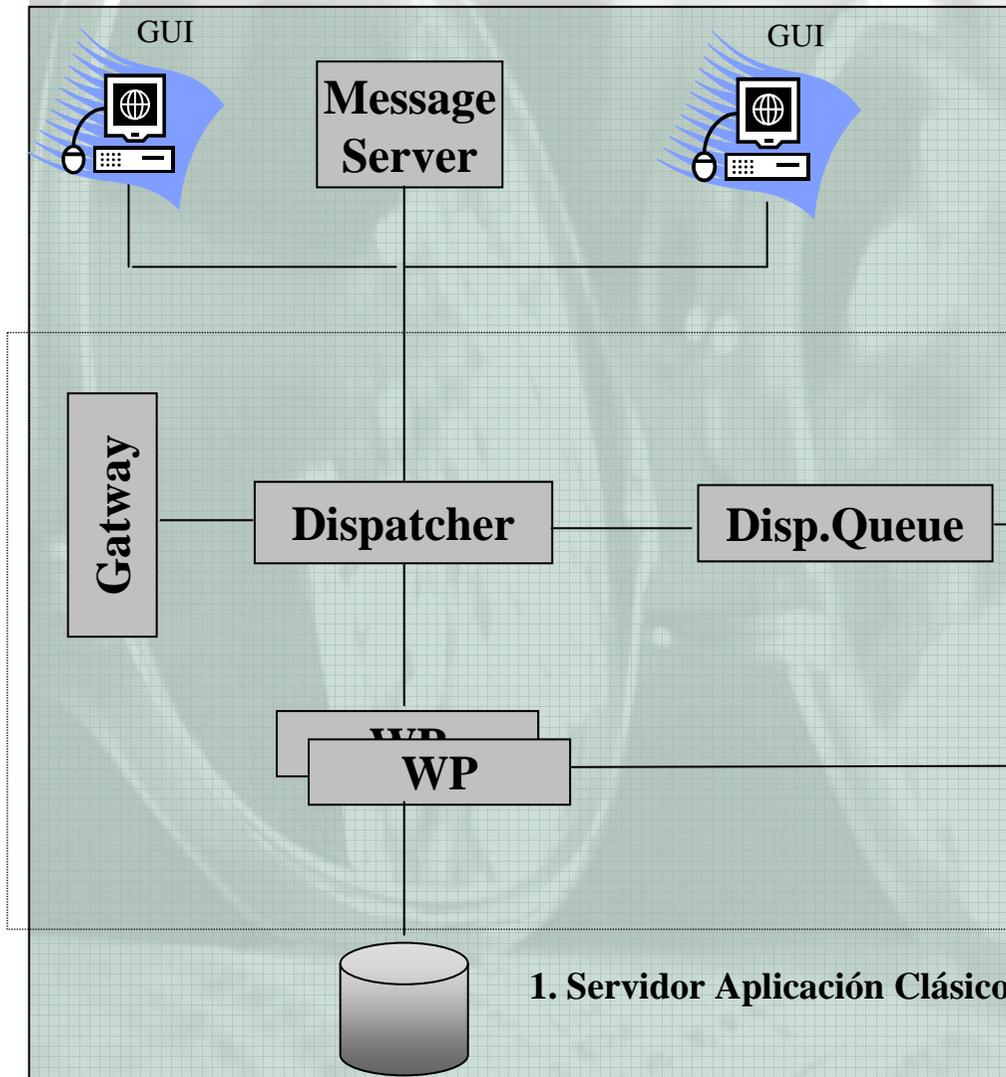


Programación de BSP's en WAS

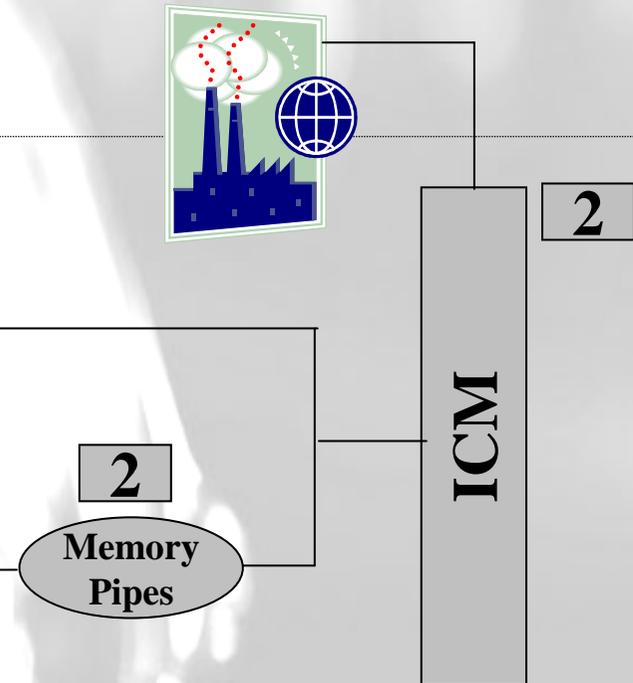
Arquitectura de WAS (I)



Arquitectura de WAS (II)



2. SAP Web Application Server



Web Application Server vs. ITS

ITS: Interface between **SAP R/3** (today: mySAP Enterprise Resource Planning) and the Internet. The Internet Transaction Server (**ITS**) allows Internet and intranet users to communicate directly with **SAP R/3** by running standard transactions, function modules, and reports as Internet Application Components (IACs).

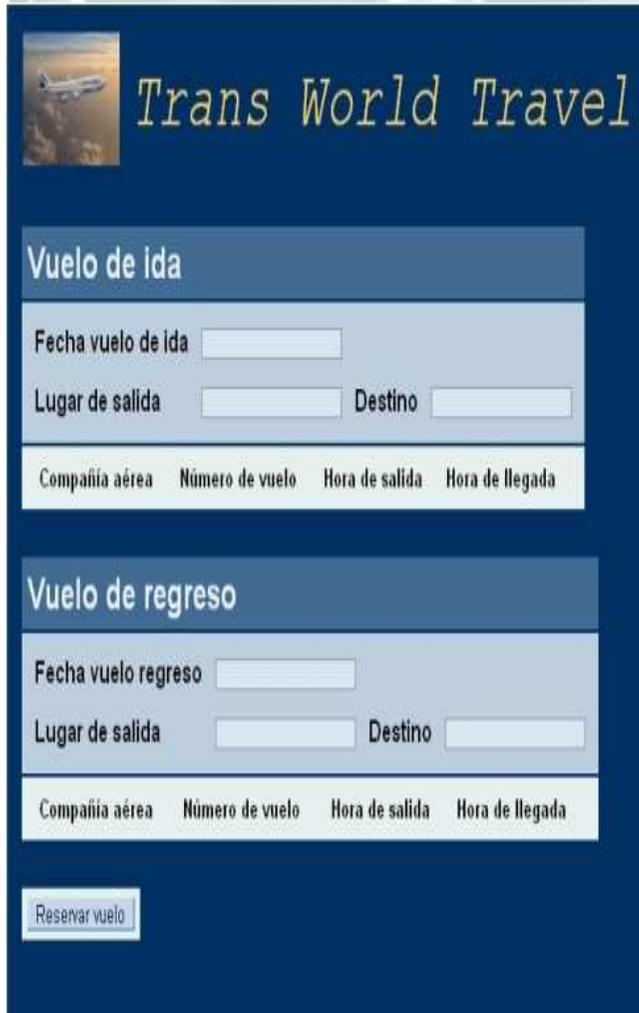
WAS

- Orientado a aplicaciones Web en Internet/Intranet que necesitan Acceso a R/3.
- Desarrollo de aplicaciones a medida.
- Aplicaciones de Internet con gran cantidad de accesos (catálogos de productos, comercio electrónico, portales, etc)
- Arquitectura sencilla y escalable.
- Servidor Web incorporado en R3.
- Funciona en todos los sistemas operativos soportados por SAP.

ITS

- Orientado para proporcionar acceso vía Internet a sistemas R/3 ya existentes.
- SAPGUI for HTML, Workplace, IAC's, etc.
- Traducción de pantallas R/3 a HTML
- Arquitectura complicada (WGate, AGate)
- Necesita Windows NT para AGate.

Business Server Pages (BSP)



 **Trans World Travel**

Vuelo de ida

Fecha vuelo de ida

Lugar de salida Destino

Compañía aérea Número de vuelo Hora de salida Hora de llegada

Vuelo de regreso

Fecha vuelo regreso

Lugar de salida Destino

Compañía aérea Número de vuelo Hora de salida Hora de llegada

- Es una aplicación WEB funcionalmente completa, equivalente a cualquier transacción clásica en SAP R/3.
- Como protocolo de comunicación de red se utiliza HTTP y HTTPS, lo que significa que sigue siendo válida la estructura de seguridad con firewalls y proxies.
- El modelo de programación de las BSP's es muy similar a los modelos de programación Java Server pages (JSP's) y Active Server Pages (ASP's), aunque ha mejorado la estructura relativa a la interfaz de usuario y a la lógica de negocio.

Componentes de una aplicación BSP (I)

PAGINAS BSP

- Representan la interfaz del usuario (presentación y definición del flujo).
- La parte de presentación incorpora normalmente código ABAP o JavaScript que se inserta en la estructura HTML mediante directivas especiales.
- Además pueden definirse manejadores de eventos y parámetros para cada página.

OBJETOS MIME

- Objetos como imágenes, iconos, hojas de estilo, archivos de audio o video....

TEMAS

- Contenedor de objetos MIME que se utiliza para ajustar la apariencia de la aplicación BSP.

CLASE DE APLICACIÓN

- Clase Global que contiene la lógica del negocio de la aplicación BSP.

Componentes de una aplicación BSP (II)

Clase de Aplicación

Business Server Pages

Atributos de Administración

Event Handlers

Layout

Atributos de Página

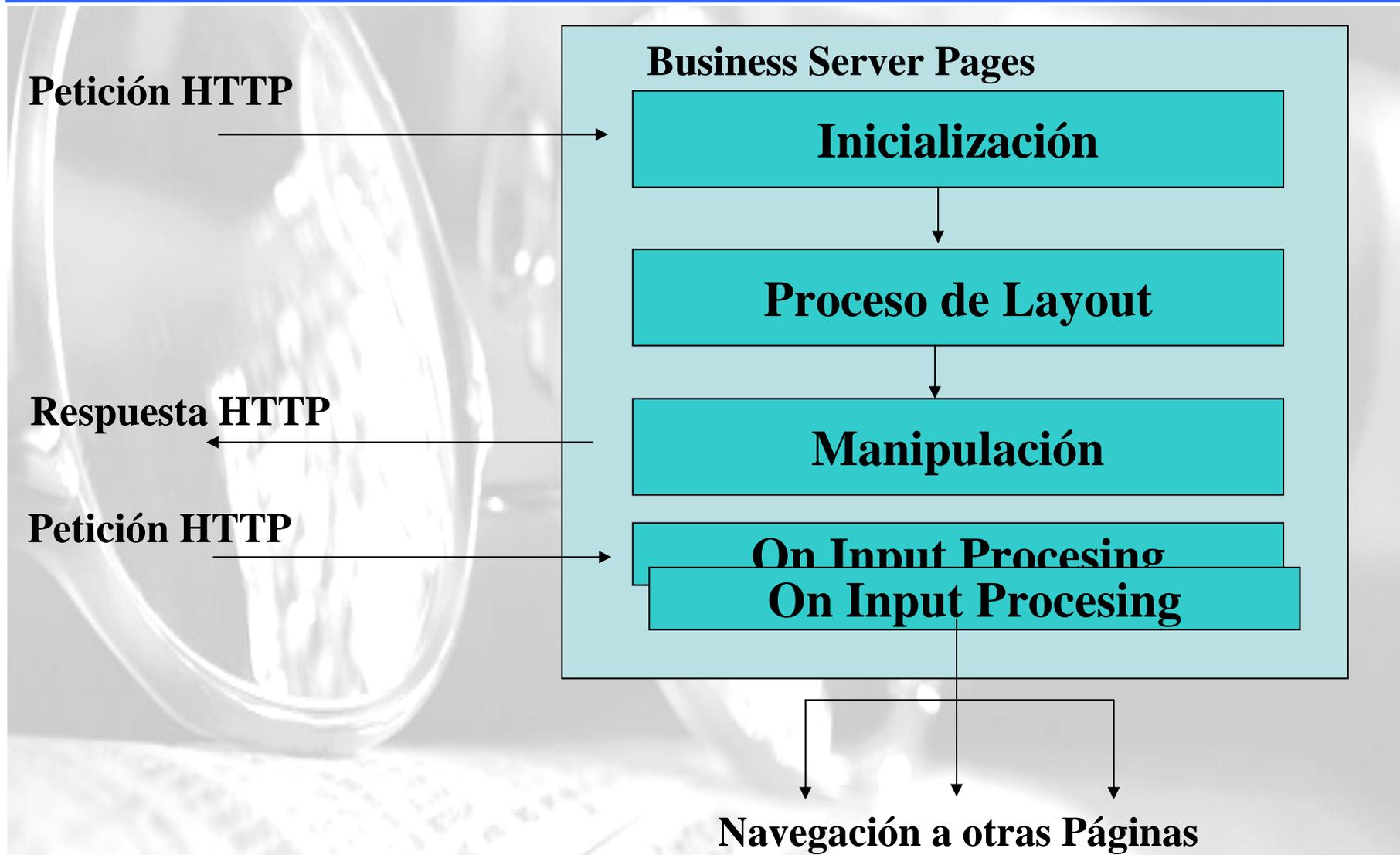
Fragmentos de Página

Objetos MIME

Estructura de Navegación

Temas

Flujo de Procesamiento de una BSP



Componentes de una página BSP

- **Descripción**
Business Server Pages (BSPs) son páginas de HTML que contienen la lógica de la Presentación y de la aplicación.
- **Layout**
Estructura física de la página BSP. Es una combinación de código estático en HTML (o XML) y scripts en ABAP o JavaScript.
- **Atributos**
Sirven para almacenar datos y hacerlos accesibles tanto a nivel de presentación como en la gestión de eventos.
- **Event Handlers**
Código ABAP que implementa la obtención de datos y el procesamiento de las acciones del usuario. Permite la separación del código estático y el dinámico en una página

Atributos de una página BSP

Descripción

Los atributos de página son parámetros declarados explícitamente para una página. Sirven para almacenar datos y hacerlos accesibles tanto a Nivel de presentación como en la gestión de eventos.

Atributos automáticos

Si un atributo de página se marca como automático, este obtiene su valor automáticamente de la URL o bien del proceso de navegación a partir de otras páginas. Sólo se tiene que declarar para una página. Si otra página utilizara un atributo con el mismo nombre el parámetro obtendría el mismo valor en tiempo de ejecución.

Ejemplo

Se ha definido FLIGHT como atributo automático para una página asignándole un valor:

```
navigation -> set_parameter(name = 'FLIGHT' value = 'LH' )
```

Si utilizásemos un parámetro del mismo nombre en otra página, este tendría el valor 'LH' en tiempo de ejecución.

Definición de tipos

```
types: begin of tipotabla,  
    dimension type int4,      "numero registros tabla  
    numeroregistros type int4, "numero registros por pantalla  
    posiciontabla type int4,  
    posicionseleccionada type int4, "posición seleccionada  
    contador type int4,      "contador de impresion tabla  
    numeropaginas type int4,  "numero de páginas tabla  
    paginaactual type int4,   "numero de página ACTUAL  
    modificable type char1,   "si se puede modificar el registro  
end of tipotabla.  
  
types: begin of tipofiltro,  
    anticipo type ZDOCANT,  
    fechaini type ZFECCON,  
    fechafin type ZFECCON,  
    status type ZESTAD,  
end of tipofiltro.  
  
types: begin of tipotablaseleccion, "seleccion de registros  
    registro type int4,  
end of tipotablaseleccion.  
types: tablasel type standard table of tipotablaseleccion.
```

Event Handlers (I)

- Código ABAP que implementa la obtención de datos y el procesamiento de las acciones del usuario.
- Pueden acceder a los atributos de página en modo de lectura y de escritura, además de tener acceso a distintos objetos globales predefinidos.
- Los gestores de eventos no deben incorporar ningún elemento de negocio, ya que para ello tenemos la clase de aplicación.
- WAS tiene predefinido los siguientes gestores de eventos, aunque pueden definirse eventos propios:

OnCreate	Cuando se crea la página por primera vez
OnRequest	Cuando se solicita la página
OnIniatilization	Obtiene la información de la BBDD.
OnManipulation	Manipulación del HTTP antes de devolverse al cliente
OnInputProcessing	Comprueba y verifica la interacción con el usuario. Navegación.

Event handlers (II)

OnCreate

Este evento se dispara cuando la página se llama por vez primera.

- Inicializa las partes del objeto página que se crean una única vez y que se utilizarán durante el resto de la vida del objeto, por ejemplo:
`create object shopbasket`
- Su principal aplicación es en BSP en las que debe mantenerse el estado. En este caso, si ejecutásemos la página una segunda vez, la instancia de la permanecería constante y no se ejecutaría OnCreate una segunda vez. (Sólo para páginas con estado)

OnRequest

Este evento se dispara cuando la pagina recibe una petición:

- Se llama a la URL desde un navegador.
- El usuario introduce datos en una página.

Event handlers (III)

OnInitiaization

Este evento se dispara después de OnRequest.

- Obtiene datos de la BBDD y los almacena en los atributos de la página para que resulten accesibles en la creación de la página y para el resto de los gestores de eventos

OnInputProcessing

Este evento se dispara cada vez que el usuario interactúa con el sistema.

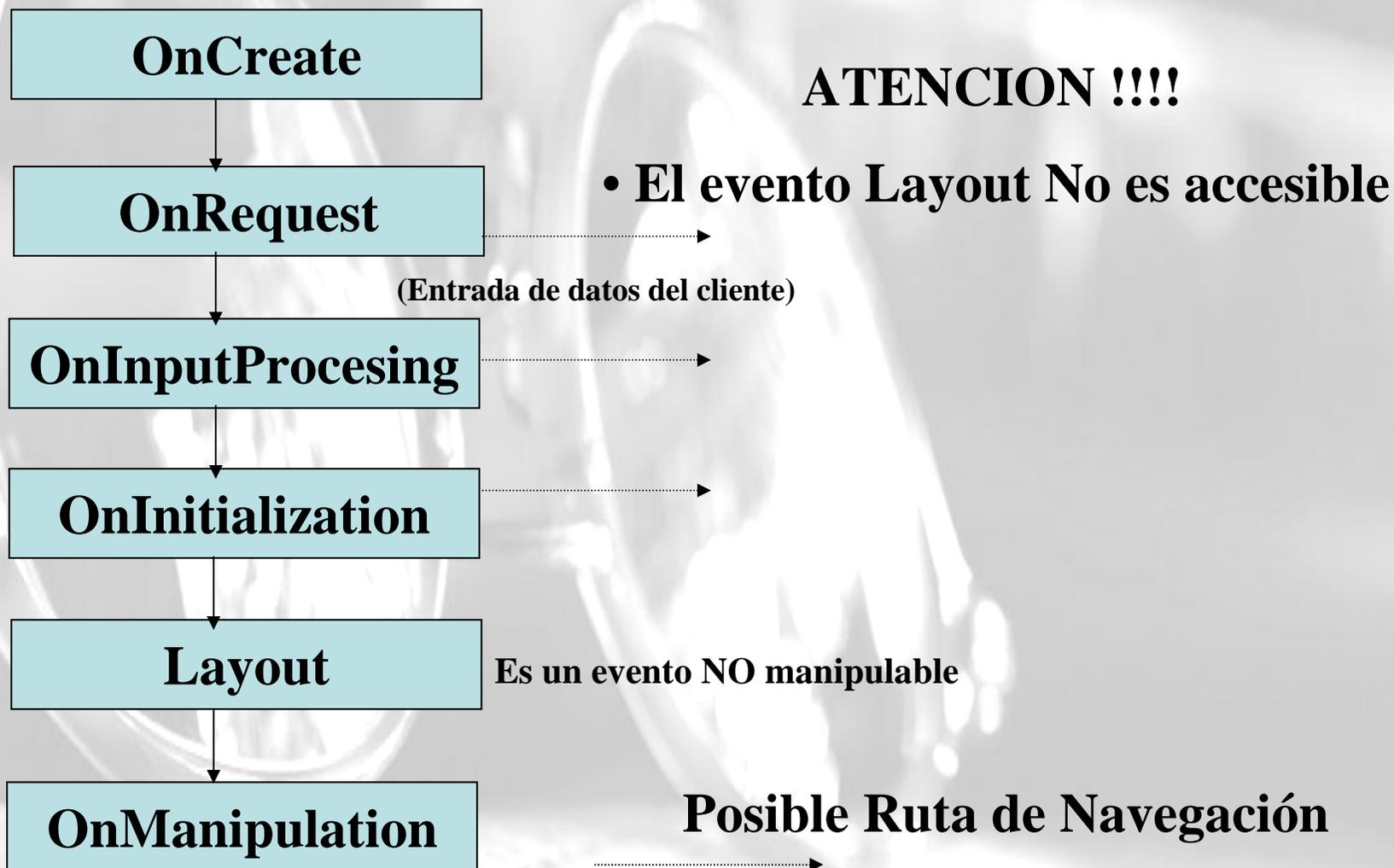
- Se utiliza para comprobar las entradas y dirigir la navegación.

OnManipulation

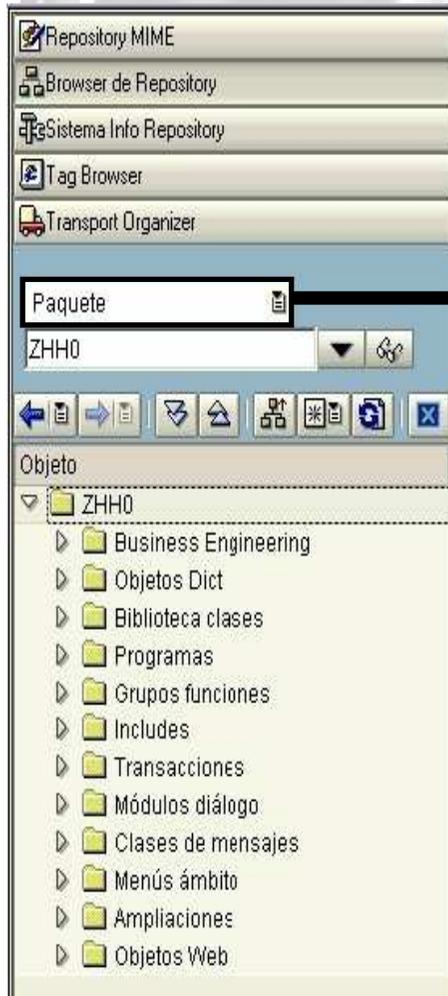
Se dispara después de que se hayan generado los componentes embebidos en la página.

- Se utiliza para manipular el datastream de HTTP después de haberlo enviado.

Flujo de la gestión de eventos



Web Application Builder (I)



Aplicación BSP

Clase/Interface

Objetos MIME

Business Server Pages

Atributos de Administración

Event Handlers

Layout

Atributos de Página

Fragmentos de Página

Clase de Aplicación

Web Application Builder

Herramienta	Características
Editor BSP	<ul style="list-style-type: none">√ Previsualización de las páginas.√ Editor básico de scripts.√ Editor de código HTML (drag & drop)√ Comprueba la sintáxis.√ Soporte para herramientas de desarrollo de terceros (WebDav).
Depurador ABAP	<ul style="list-style-type: none">√ Depuración integrada de scripts ABAP y JavaScript en el servidor.
Librería de TAGs	<ul style="list-style-type: none">√ Librería de TAGs para estándares como HTML, WML y XHTML.
Repositorio MIME	<ul style="list-style-type: none">√ Almacena imágenes, íconos, páginas de estilo, etc...√ Accesible desde herramientas de desarrollo con WebDav.√ Integrado con el sistema de transporte (CTS)
Editor de TEMAS	<ul style="list-style-type: none">√ Permite la personalización de las aplicaciones WAS.

Depuración de código por HTTP



```
ABAP_DEBUGGER
Watchpoint
Campos: Tabla Break
Programa ctrl CL_023UDS07
Cód fuente de CL_023UDS07
METHOD _ONINPUTPROCESSING
objamodifyf->employ
* objamodifyf->langua

clear mensaje.

***** RECOGEMOS LOS
cuenta = request->get
➔ [BP] digito = request->get
entidad = request->ge
sucursal = request->g

cuenta_6V = request->
digito_6V = request->
entidad_6V = request-
sucursal_6V = request
```

Descripción

Permite depurar páginas BSP desde WEB para un determinado usuario.

Flujos de Proceso: Debbing.

1. Seleccionar la aplicación BSP en la transacción SE80.
2. Seleccionar en el menu: Utilidades → Breakpoints → (Des)Activar para usuario HTTP.
3. Introducir el nombre de usuario.
4. Ejecutar la aplicación. La depuración se realiza de la misma forma que en el ABAP IV tradicional.

Para desactivar la depuración bastará con seguir los mismos pasos.

Ejemplo I – Creación de una aplicación BSP

Flujo de proceso

Existen seis pasos básicos involucrados en la creación de una aplicación BSP.

1. Definición de una nueva aplicación BSP.
2. Añadir parámetros a la página.
3. Crear la apariencia de la página utilizando HTML.
4. Incluir los scripts de servidor.
5. Añadir los manejadores de eventos.
6. Añadir los toques finales.

Ejemplo I – Definir una nueva aplicación BSP

Flujo del proceso.

Para comenzar vamos a crear un pequeño listado:

1. Entrar en la transacción SE80.
2. Seleccionar “BSP Application” en la lista de tipos de objeto e introducir un nombre para la nueva aplicación. Ahora se pedirán los datos típicos de descripción, paquete de desarrollo, orden de transporte... Una vez creada la aplicación, la navegación se efectúa como con cualquier otro tipo de objeto tradicional.
3. Crear la primera página de la aplicación BSP. Para ello, seleccionar el nodo de la aplicación y Crear → Página.
4. Introducir un nombre para la página (display.htm) y grabar. Ya tenemos creada la página.

Ejemplo I – Definir una nueva aplicación BSP

The screenshot displays the SAP Web Application Builder interface for configuring a new BSP application. The title bar shows 'Web Application Builder: Modificar aplicación BSP'. The main window is divided into several sections:

- Left Panel:** Contains navigation tools and a tree view of the application structure. The tree shows a folder 'zbsp_flight' containing a sub-folder 'Páginas con lógica proceso' with files 'display.htm' and 'search.htm'.
- Top Bar:** Includes the SAP logo and a menu with options: 'Aplicación BSP', 'Tratar', 'Pasara', 'Utilidades', 'Entorno', 'Sistema', and 'Ayuda'.
- Main Content Area:** Displays the configuration for the application 'zbsp_flight'. It has tabs for 'Atributos' and 'Navegación'. The 'Atributos' tab is active, showing the following details:
 - Aplicación BSP: zbsp_flight (Estado: Activo/revisado)
 - Descripción breve: Prueba
 - Creado por: SRGEN2, Fecha creación: 02.09.2004
 - Modificado por: SRGEN2, Fecha modificación: 02.09.2004
 - Paquete: \$TMP
 - Idioma maestro: ES
 - Nombre interno: ZBSP_FLIGHT
 - Imagen inicial BSP: search.htm
 - Clase de aplicación: ZWEB_LIQ_SESSION
 - Tema: (empty field)
 - Con estado
 - Soporta integración portal
- Bottom Bar:** Shows the user 'SRGEN2' and session information 'espux008 INS'.

Ejemplo I – Añadir parámetros a la página

Flujo de Proceso.

Para añadir parámetros a la página:

1. Seleccionar la pestaña de Atributos de Página en el editor.
2. Introducir el parámetro llamado *flights*. Este será una tabla de registros con la estructura SFLIGHT. En el tipo de dato introducimos FLIGHTTAB.
3. Introducir un segundo parámetro llamado *flight*. Este se necesita como área de trabajo. Su tipo es SFLIGHT.

Ejemplo I – Crear la interfaz del usuario en HTML

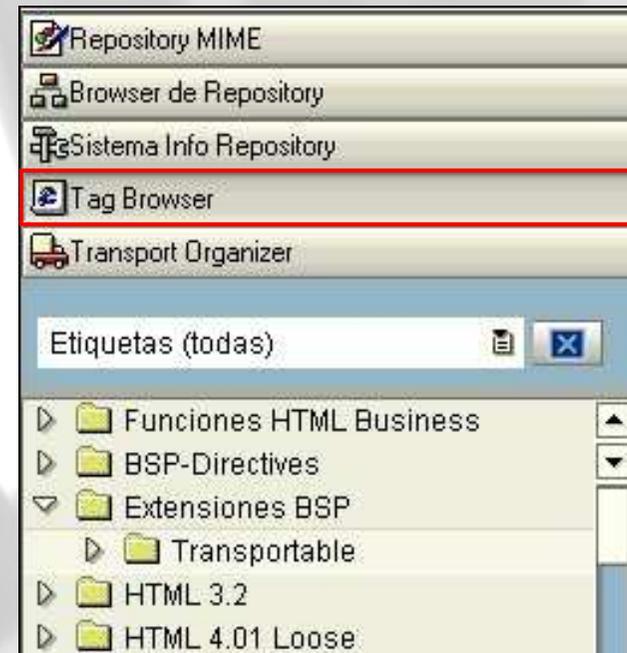
Flujo del proceso

Aquí es donde intervienen nuestros extensos conocimientos en HTML (siempre se puede utilizar la librería de TAGs de HTML en la SE80)

- 1.- Abrir la librería de TAGs
pulsando el botón de la barra de herramientas.
- 2.- Expandir la rama correspondiente al HTML 3.2.
- 3.- Como vamos a mostrar un listado,
necesitaremos un esqueleto de página
de tipo tabla (ver tablas en HTML).

Hacer Drag&Drop con los elementos necesarios.

<code><table></code>	Comienzo de tabla
<code><tr></code>	Comienzo de línea
<code><td></td></code>	Columna
<code></tr></code>	Final de línea
<code></table></code>	Final de tabla



Ejemplo I – Incluir scripts de servidor

Flujo de proceso

Para señalar el tipo de script que se va a introducir en la página se inserta en la primera línea

```
<% @page language="abap"%>
```

O bien,

```
<% @page language="JavaScript"%>
```

También podemos seguir utilizando la librería de tags pero esta vez seleccionando “BSP Directives”

Ejemplo I – Incluir scripts de servidor

Flujo de proceso

Vamos a mostrar los registros de una tabla que almacenamos en el parámetro de página flights.

```
<table>
<% Loop at flights into flight. %>
  <tr>
    <td> <% flight-carrid %> </td>
    <td> <% flight-connid %> </td>
    <td> <% flight-fldate %> </td>
  </tr>
<% Endloop. %>
</table>
```

Ejemplo I – Añadir los manejadores de eventos

Flujo de proceso

- Vamos a implementar el código que devuelve los datos de los vuelos de la tabla SFLIGHT.
- Seleccionar el TAB Event Handler.
- De la lista de gestores de eventos, seleccionar “OnInitialization”.
- Añadir el código siguiente:

Select *

From sflight

Into corresponding fields of table flights up to 100 rows.

Ahora ya podemos activar la aplicación y corregir los errores que hayamos cometido. Para probarla presionaremos F8.

Ejemplo I – Añadir los toques finales

¿Qué puede fallar en el listado ? Suele ser de utilidad añadir una Cabecera... ;-)
Volvemos a editar el HTML y añadimos

```
<table>  
<th>Carrid</th>  
<th>Connid</th>  
<th>fldate</th>  
<%Loop at flights into flight. %>  
...  
<%Endloop. %>  
</table>
```

Ejemplo I – Añadir los toques finales

Flujo del proceso

Mejoremos un poco la apariencia de nuestra página utilizando una hoja de estilo del repositorio

- Seleccionar el repositorio MIME presionando el botón correspondiente d la barra de herramientas.
- De la rama pública, elegimos la hoja de estilos deseada, sapbsp.css.
- Arrastrar el estilo a la posición correspondiente del tag<link> situado en la cabecera del documento,

```
<link rel="stylesheet" href="...sapbsp.css">
```

4. Ahora aplicamos el estilo a la tabla,

```
<table class="bspTbvStd">
```

Ejemplo I – Añadir los toques finales

Flujo del proceso

Escribamos el campo fldate en formato de fecha.

- Reemplazar el campo `<%=flight-fldate%>` en la tabla por lo siguiente:

```
<%  
    data: str type char10.  
    write flight-fldate to str .  
%>
```

Este código convierte el formato interno de fecha al formato preferido por el usuario, de acuerdo con los datos maestros de la base de datos.

Ejemplo I – Optimizando el código de servidor

```
<% data: trow type string, tstyle type string, iCont type i .
```

```
Loop ata flights into flight.
```

```
Trow = '<tr class=$0><td>$1</td><td>$2</td><td>$3</td></tr>'
```

```
iCont = sy-tabix mod 2.
```

```
If iCont eq 0.
```

```
replace '$0' into trow with ' bspTbvCellStd' .
```

```
else.
```

```
replace '$0' into trow with ' bspTbvCellAlt' .
```

```
endif.
```

```
replace '$1' into trow with ' flight-carrid.
```

```
replace '$2' into trow with ' flight-connid.
```

```
replace '$3' into trow with ' flight-fldate.
```

```
page->write( trow ) .
```

```
Endloop %>
```

Introducción al HTML(I)

Comienzo del Documento → **<html>**

Cabecera del Documento →

```
<head>  
<!--Cabecera del Documento >  
  <title>Estructura básica de un documento</title>  
</head>
```

Comentario →

Cuerpo del documento →

```
<body>  
<!--Cuerpo del Documento >  
  <h1>HTML Básico</h1>  
  <p>En este documento se muestra la estructura  
<i>básica</i> de una pagina HTML  
</body>
```

Final del documento → **</html>**

Introducción al HTML: cabecera (II)

```
<html>
```

```
<head>
```

```
  <title>estructura básica de un documento</title>
```

```
  <meta http-equiv = "Expires" content = "tue, 04 Dec 2002 21:29:02 GMT">
```

```
  <meta http-equiv = "Refresh" content = "5,http://www.sample.com/next.html">
```

```
  <meta http-equiv = "Keywords" content="HTML,introduccion ">
```

```
  <meta http-equiv = "Reply-to" content = zuriburi@hotmail.com>
```

```
  <meta name = "author" content="German Mesa">
```

```
  <linkrel="styleheet" href = "libro.css">
```

```
</head>
```

Navegación en HTML

- Un hyperlink es un texto subrayado y coloreado, o gráfico, que activa una URL para descargar y mostrar otro archivo. La sintáxis es la siguiente

Para un texto:

```
<A HREF="http://www.sap.com">SAP Home Page</A>
```

Para una imagen:

```
<A HREF="http://www.sap.com"><IMG SRC = "images/sap.gif">SAPHome</A>
```

- En el caso de que quisiésemos especificar una posición concreta dentro del documento podríamos utilizar un anchor con un atributo de tipo NAME.

default.html

```
Para más información <A HREF="info.html#suplementaria">InfoPage</A>
```

Info.html

```
<A NAME="suplementaria">Información suplementaria</A>
```

Tablas en HTML (I)

TABLA

```
<table width="100%"border="1">
```

```
<tr>
```

```
<td>Columna 1</td>
```

```
<td>Columna 2</td>
```

```
</tr>
```

```
</table>
```

CABECERA

```
<thead>
```

```
<tr>
```

```
<td>columna 1</td>
```

```
<td>columna 2</td>
```

```
</tr>
```

```
</thead>
```

Comienzo de tabla

Comienzo de fila

Columnas

Final de fila

Final de tabla

Comienzo de cabecera

Comienzo de fila

Columnas

Final de fila

Final de cabecera

Tablas en HTML (II)

Cuerpo

```
<tbody width="100%" border="1">
  <tr>
    <td colspan="2">Columna 1 y 2</td>
  </tr>
  <tr>
    <td rowspan="2">Columna 1: Fila 2 y 3</td>
    <td> Columna 2 </td>
  </tr>
  <tr>
    <td>columna 1</td>
    <td>columna 2</td>
  </tr>
</tbody>
```

Comienzo de cuerpo

Columnas 1 y 2 se juntan

La columna 1 es común
para las dos filas
siguientes

Formularios en HTML

- Un formulario es un contenedor de controles desde donde el usuario puede introducir datos en una página Web.
- Un Formulario esta delimitado por la etiqueta `<FORM>... </FROM>`.

Parámetro	Valor	Descripción
Method	Get	Protocolo utilizado para enviar los datos el servidor El navegador concatena todos los parámetros al final de la cadena URL.
	Post	No aparece en la lista de parámetros al final de la URL
Action	*	Indica la acción que debe realizarse una vez se envíen los datos al servidor.

Formularios en HTML (II)

- Para introducir datos en el formulario, utilizamos el tag <INPUT>

Parámetro	Valor	Descripción
Type		Tipo de introducción de datos por el usuario
	Text	Campo de texto
	Checkbox	Checkbox
	Radio	Radio botón.
	Button	Botón
	Hidden	No se muestra al usuario, y no puede por tanto modificarlo.
	Submit	El navegador junta todos los parámetros y los reenvía al servidor.

Formularios en HTML (III)

Campo oculto

```
<input name="hiddenField" type="hidden" value="50843415Y">
```

Campo de texto normal

```
<input name="textfield" type="text" value="Usuario" size="20" maxlength="10">
```

Campo de texto para contraseña

```
<input name="password" type="password" id="password" size="20" maxlength="10">
```

Checkbox

```
<input type="checkbox" name="checkbox" value="checkbox">
```

Formularios en HTML (IV)

Radio Button

```
<input type="radio" name="radio" value="opción">
```

Grupo de opciones

```
<label><input type="radio" name="Grupo _Opciones" value="1">Primera</label>  
<label><input type="radio" name="Grupo _Opciones" value="2">Segunda</label>  
<label><input type="radio" name="Grupo _Opciones" value="3">Tercera</label>
```

Botones

```
<Input type=" submit" name="Submit" value="Enviar">  
<Input type=" reset" name="Restablecer" id="Restablecer" value="Restablecer">
```

Formularios en HTML (V)

```
<FORM METHOD="POST" NAME="insUser">
<TABLE>
<TR><TD>Nombre:<input name="nombre" type="TEXT" id="nombre" value="" size="30"/>
    </TD></TR>
<TR><TD>País:
    <select name="pais" size="1" id="pais">
    <option value="DE" selected>Alemania</option>
    <option value="ES">Españ<a>/option>
</TD></TR>
<TR><TD>Contraseña<INPUT NAME="pass" TYPE="password" size"10" maxlength="10"/>
    </TD></TR>
<TR><TD><input name="Restablecer" type="reset" value="Restablecer">
    </TD></TR>
<TR><TD><input name="Submit" type="submit" id="Submit" value="Enviar" width="100%">
    </TD></TR>
</TABLE>
</FORM>
```

Estilo de la aplicación (I)

Las hojas de estilo se utilizan para controlar la apariencia global del sitio Web, de tal forma que si quisiera cambiar el aspecto, sólo se debería modificar la hoja.

- Una hoja de estilos es un archivo de texto compuesto por **reglas**:
- ```
H3{font-family:Arial;
```
- ```
font-style:italic;
```
- ```
color:green }
```
- Cada regla comienza por un selector, normalmente una etiqueta HTML, y declaraciones, que describen cómo deben aparecer estos elementos.
- También pueden utilizarse **clases** como selectores, que no se asocian a un elemento HTML:

```
.greenitalic{ font-family: Arial;
```
- ```
font-style:italic;
```
- ```
color:green }
```
- Para aplicar una clase a un tag HTML, se utiliza el atributo **class**:
- ```
<H3 CLASS="greenitalic">este estilo es greenitalic<H3>
```

Estilo de la aplicación (II)

Existen varias formas en las que se puede aplicar el estilo a una hoja:

- Crear los documentos HTML utilizando el atributo style. No se recomienda por que se pierde el beneficio de mantener el contenido separado de la apariencia.
- Utilizar un bloque de estilos en la sección de cabecera del documento HTML. Un bloque de estilos se compone de un conjunto de reglas de estilo entre dos etiquetas <STYLE></STYLE>. Es mejor que el anterior pero no se recomienda.

```
<STYLE>  
    BODY { color: navy;}  
    H1 {font-size: 12pt; }  
</STYLE>
```

Ligar una página de estilo al documento utilizando la etiqueta <LINK> en la cabecera del documento HTML.

```
<head>  
    <link rel="stylesheet" type="text/css" href="style.css">  
</head>
```

AYUDAS!!!

Links manuales HTML:

<http://www.programacion.com>

<http://www.desarrolloweb.com>

Link hoja de estilos:

<http://www.w3schools.com/css/default.asp>

Link SAP SDN:

<https://www.sdn.sap.com/sdn/weblogs/sdn?blog=/pub/wlg/542>

Protege tu marca con el .es
Registrarte tu dominio
tu dominio con acens

Portada | Monotemáticos | Secciones | Comunidad | Servicios gratuitos | Servicios profesionales

Monotemáticos + HTML a fondo + Promoción de webs + ASP a fondo + PHP a fondo	Novedades Enviar formulario con PHP 05/11/2004 - Enviar desde un formulario informacion en un archivo por mail con PHP. El lenguaje XBRL	Contenidos por temas Internet Organizaciones, Medios, ... Diseño Guías de diseño, Programación HTML, Javascript, ASP, PHP, Java, .NET, ... Marketing Promoción, Dinero,	Añadir URL
---	--	--	------------

Search hundreds of web hosts with solutions as low as
Free setup, domain and 800 support | 40 GB transfer and money back guarantee | Linux hosting from \$9/month

CSS Tutorial

Save a lot of work with CSS!
In our CSS tutorial you will learn how to use control the style and layout of multiple Web once.
Start learning CSS!

CSS Examples

Learn by 70 examples! With our editor, you can edit CSS, and click on button to view the result.

CSS Quiz Test

Test your CSS skills at W3Schools!

- CSS HOME
- CSS Home
- CSS Introduction
- CSS Syntax
- CSS How To
- CSS Background
- CSS Text
- CSS Font
- CSS Border
- CSS Margin
- CSS Padding
- CSS List
- CSS Advanced
- CSS Dimension
- CSS Classification
- CSS Positioning
- CSS Pseudo-class
- CSS Pseudo-element
- CSS Media Types
- CSS Examples/Quiz
- CSS Examples
- CSS Quiz
- CSS Exam
- References

SAP DEVELOPER

Member Login

User ID or Email
Password
Remember me
Forgot your password? Log in

Search

SDN Content

Developer Areas

- SAP NetWeaver Platform
- Enterprise Portal
- Knowledge Management

Buscador

En nuestro sitio

Secciones

- Noticias
- Cursos
- Articulos
- Foros
- Código fuente
- Direcciones
- Formación

Otras zonas

- ASP en castellano
- Bases de datos en castellano
- Java en castellano
- PHP en castellano

Otras zonas

- HTML
- JavaScript
- XML
- y mucho más

BSP Trouble Shooting: Getting Help

Brian McKellar SAP
Jun. 11, 2004 01:34 PM
[Permalink](#)

Every other day, and sometimes it feels like every other minute, the simple question comes up: "How to do X?", usually with the "Urgent!" or "????". Most of these questions, when reading between the lines, also contain the additional qualifier "I have not read the documentation". In a way, this is understandable. Reading documentation is hard work and boring. However, we have to look at it also from the perspective of the person asking the question. It is frustrating to feed people every day with a little spoon. Especially after a hard day's work. This Weblog attempts to point to interesting and relevant sources of information. It is not an answer to any question; no, it is the

Overview of the More Important Bits and Pieces

Business Server Pages (BSP) is just the tip of the iceberg. It builds on the foundations of many other people's hard work. By the t

Utilización de Editores externos

Requisitos

- Pueden utilizarse herramientas externas, esto es, entornos de desarrollo específicos para el desarrollo en Web para implementar la apariencia de las páginas BSP. Esto requiere que la herramienta soporte el protocolo WebDav y que pueda funcionar como un cliente WebDav.
- El servidor WAS contiene un servicio WebDav que le permite funcionar como servidor WebDav.
- WebDav es una extensión del protocolo HTTP que proporciona una infraestructura para editar documentos en entorno de trabajo Web distribuidos.

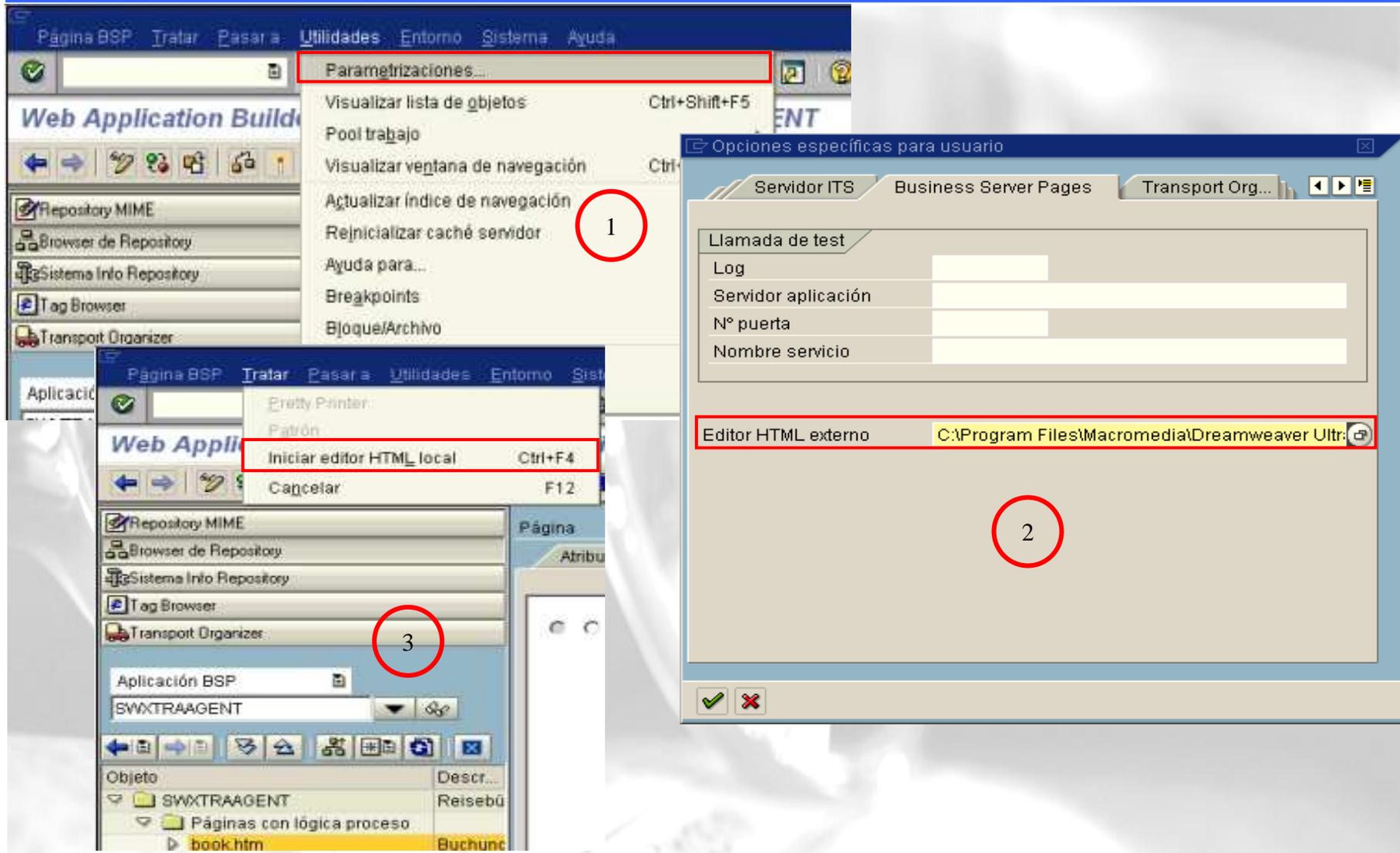
Procedimiento

En la transacción SE80:

1. Seleccionar en el menú Utilidades->Settings.
2. Seleccionar el TAB `Business Server Pages´ e introducir un editor, p.ej.Dreamweaver.

Para utilizarlo marcar la página y editar con la opción `Ejecutar editor local´.

Utilización de Editores externos



Ejemplo II- Creación de un formulario de datos.

Flujo del proceso

Vamos a crear una página inicial desde la que llamar al listado que hemos creado anteriormente.

- 1.** Entrar en la transacción SE80 y pulsar el botón Repository Browser de la barra de herramientas.
- 2.** Abrir el menú de contexto y añadir otra página llamada search.htm a la aplicación BSP.
- 3.** Crear un formulario donde se solicite la compañía aérea y con un botón para enviar la petición.

Ejemplo II- Creación de un formulario de datos.

El código quedaría como sigue:

```
<form method="POST">  
  Compañía: <input name="carrid"/>  
  <input value="Buscar..." type="SUBMIT" name="OnInputProcessing (Select)"/>  
</form<
```

Cuando se pulsa el botón Buscar, se transfiere el control al WAS, que llama al evento de la página OnInputProcessing. Para permitir selecciones múltiples en el manejador de eventos, se puede añadir una cadena que identifique el botón que se ha seleccionado.

Antes de implementar el código que procese la selección del usuario, debemos cambiar un poco la página display.htm para que acepte el parámetro de filtro

Ejemplo II- Creación de un formulario de datos.

Hay que añadir un nuevo parámetro que contenga el identificador de la compañía:

1. Seleccionar el tab de “Atributos de Página” de la página display.htm.
2. Añadir un nuevo parámetro, carrid, de tipo sflight-carrid. Este parámetro será de tipo automático, lo que asegura que se cargará con el valor adecuado.
3. Modificar el Event Handler que carga los datos iniciales para añadirle el parámetro, i.e.

```
select *  
form sflight  
into corresponding fields of table flights up to 100 rows  
where carrid = carrid.
```

Ejemplo II- Creación de un formulario de datos.

Ahora es el momento de añadir la estructura de navegación en la página search.htm para que incluya el parámetro:

1. Abrir la página search.htm.
2. Seleccionar el evento OnINputProcessing.
3. Introducir el código siguiente:

```
data: carrid type sflight-carrid.
```

```
carrid = request ->get_form_field ( name = 'CARRID ').
```

```
navigation->set_parameter( name = 'Carrid' value = carrid).
```

```
navigation->goto_page( 'display.htm').
```

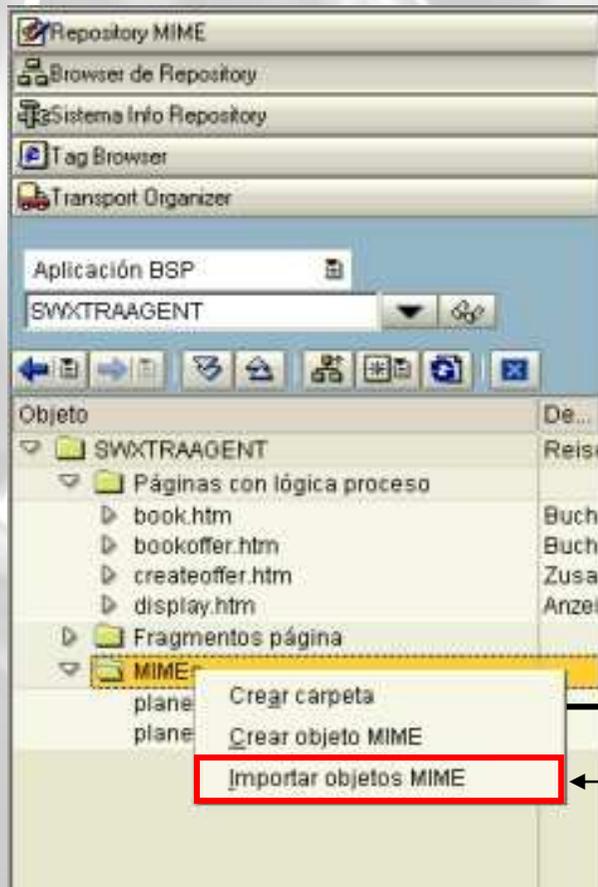
4. Activar y ejecutar la aplicación

Repositorio MIME



- **Transacción SE80 o bien SO2_MIME_REPOSITORY**
- **Se utiliza para almacenar todos los objetos MIME:páginas de estilo, gráficos, iconos.,fichero XML,...**
- **Los objetos MIME se lamacenan en la base de datos de SAP y pueden utilizarse en las aplicaciones BSP.**
- **Los objetos MIME estan asociados a una orden de transporte.**
- **El sistema crea automáticamente un directorio en el repositorio asociado a la BSP deonde se almacenan todos los objetos asociados a la aplicación.**
- **Existe un directorio público donde se almacenan todos aquellos MIMEs utilizados en varias aplicaciones.**

Importar y borrar MIMEs



Sólo un objeto cada vez



Importación masiva de MIMEs utilizando WebDAV

Descripción

El contenido del repositorio MIME constituye un sistema de archivos virtual que resulta visible a todos los clientes con WebDAV (por ejemplo Windows NT Explorer).

Procedimiento

En Windows NT:

1. Abrir el NT Explorer.
2. Anadir un nuevo directorio Web utilizando el Wizard situado en el directorio Web Folders. La ubicación del directorio web es:

http://<SAP WebAS>:<Puero>/sap/bc/bsp_dev

Nota: Si estuviésemos en Windows 2000 mapearíamos una unidad de red seleccionando la posibilidad para crear un acceso directo al directorio Web.

Atención: Mucho cuidado con sobrescribir cualquier objeto de la carpeta PUBLIC del repositorio MIME. Al estar todas las aplicaciones referidas a esta carpeta cualquier cambio puede resultar catastrófico

Caché del servidor: Eliminar elementos de la caché

Descripción

- Todos aquellos objetos de uso habitual se pueden almacenar en una caché. Este mecanismo optimiza el rendimiento del sistema.
- Cuando se vuelve a solicitar la página, esta se envía directamente desde el ICM al cliente, sin tener que generarla de nuevo.
- Todos los objetos MIME se guardan en la caché por defecto.
- Problema para el desarrollo: una vez que los objetos MIME se han cargado en caché el ICM no vuelve a buscarlos al repositorio, por lo que cualquier variación que se haga en ello no resultará visible.

Flujo de Proceso

1. Entrar en la transacción SMICM.
2. Seleccionar en el menú: Gestión->Caché del servidor->Visualizar.
3. Seleccionar el elemento que se desea eliminar de la caché
4. Eliminar utilizando el botón Borrar.

Ejemplo III- Repositorio MIME

1. En la transacción SE80, pulsar el botón “ Repositorio MIME” de la barra de herramientas.
2. En el árbol de aplicaciones, seleccionar el contenedor con el nombre de nuestra aplicación y abrir el menú de contexto con el botón derecho del ratón. Vamos a crear un nuevo directorio.
3. Especificar un nombre (imágenes) y grabar.
4. Situándonos en el directorio, botón derecho, elegimos la opción Importar objeto MIME.

Nota:

Si no aparece el botón Repositorio MIME en la pantalla , navegar a “ Entorno->Settings” y seleccionar la pestaña “ Workbench (General)”. Seleccionar “ Repositorio MIME” de la lista de herramientas que aparecen.

Ejemplo III- Repositorio MIME

5. Seleccionar un archivo de imagen del sistema de archivos y pulsar Enter. Ahora la imagen se ha descargado al repositorio MIME y aparece en el directorio.
6. Si hacemos doble clic en la imagen podremos previsualizarla.
7. Abrir la página de búsqueda `search.htm` y seleccionar la pestaña “Disposición”
8. Ahora insertaremos en la página una imagen utilizando el tag ``. Insertamos el siguiente texto en la posición del formulario que deseemos:
``
9. Finalmente, sólo nos queda arrastrar la imagen del repositorio MIME a la posición entre comillas, con lo que nos queda algo parecido a:
``

Ejemplo III- Importación masiva de MIMEs

Tras la experiencia traumática de la inserción de una sola imagen en el formulario, nos encontramos ahora preparados para apreciar la importación masiva de objetos MIME.

1. Abrir el explorador de archivos.
2. Mapear el directorio del repositorio de objetos tal y como se describe en la transparencia “ Importación masiva de MIMEs “
3. Agrupar todas aquellas imágenes que necesitemos en la aplicación en una carpeta en el sistema de archivos de PC.
4. Marcar todas aquellas imágenes que deseemos copiar en el repositorio MIME.
5. Copiar y pegar las imágenes en el repositorio MIME.

Nota: Para poder utilizar este procedimiento, necesitaremos tener los permisos adecuados para nuestro usuario WAS. Todos los objetos MIME quedarán agrupados en una orden de transporte.

Fragmentos de página

Descripción

Todas aquellas partes de la página que va a repetirse a lo largo de la aplicación, como cabeceras, logos, estructuras de navegación... pueden almacenarse en fragmentos de la página.

Flujo del proceso

Desde la transacción SE80.

1. Seleccionar la aplicación BSP
2. Pulsando el botón derecho del ratón, seleccionar Crear->Página HTML.
3. Introducir el nombre del fragmento y marcar el check box “ Fragmento de Página ”.
4. Introducir el código HTML que conformará el fragmento de la página.
5. Grabar y activar.

Estos fragmentos, se incluyen en la páginas introduciendo la directiva BSP:

```
<% @include file="fragmento.htm"% >
```

Ejemplo IV

Fragmentos de página

Vamos a utilizar tablas e imágenes para crear pie de página sencillo para nuestras páginas BSP

Desde la transacción SE80:

1. Seleccionar la aplicación BSP.
2. Pulsando el botón derecho del ratón, seleccionar Crear->Página HTML.
3. Introducir “ footer.htm” como nombre del fragmento y marcar el check box “ Fragmento de Página ”.
4. Introducir el código HTML que conformará el fragmento de la página.
5. Grabar y activar
6. Incluir el fragmento en aquellas páginas que se desee, grabarlas y activarlas.

Construir páginas utilizando herramientas externas

Descripción

Se crea la estructura de la aplicación con la transacción SE80 y se modifica la apariencia utilizando herramientas externas

Flujo del Proceso

1. Crear la estructura básica de la aplicación con la transacción SE80
2. Editar las páginas utilizando WebDAV.
3. Importar los MIMEs necesarios al Repositorio MIME de WAS
4. Implementar la apariencia de la aplicación creando las páginas necesarias.
5. Importar las páginas HTML al repositorio MIME.
6. Convertir las páginas estáticas del repositorio en BSPs, si fuese necesario.
7. Implementar los eventos de la página, definir los parámetros de página y probarla.

Creación de una estructura de navegación (I)

Descripción

Después de crear las páginas de la aplicación BSP, se determinan las transacciones entre páginas

Procedimiento

Esto puede hacerse de tres formas diferentes:

- Navegación via HTML
`Publicidad`
- Navegación via FORMS. Se especifica en el parámetro 'action' del formulario.
`<form name="logon" nethod="post" action="info.htm">`
...
`</form>`
- Navegación vía gestor de eventos.

Creación de una estructura de navegación (II)

Descripción

Proporcionando una “Navigation Request” a la transición entre dos páginas, el sistema determina automáticamente las páginas de origen y destino.

Procedimiento

1. Seleccionar la aplicación BSP en la transacción SE80
2. En el TAB “Navigation” introducir todas las transacciones posibles entre las páginas de la aplicación que vayan a ser gestionadas por la gestión de eventos.

Cada transacción debe tener .

- Página origen
- Página destino
- Nombre

Nota: Esto mismo puede hacerse desde el “Navigation Modeler”

Ejemplo V

Creación de una estructura de navegación (II)

1. Seleccionar la aplicación BSP.
2. En la pestaña Navegación seleccionar la página search.htm como página inicial
3. Seleccionar la página display.htm como página destino
4. Especificar un nombre para la petición de navegación (ONSEARCH)
5. Si ahora seleccionamos la pestaña “ Modelador de Navegación” aparecerá una representación gráfica del modelo lógico de la aplicación.
6. Finalmente hay que modificar el código de navegación de la página search.htm para hacer referencia a este cambio. En el código, cambiamos:
 navigtion->goto_page (‘dispaly.htm’)
por
 navigation->next_page (‘OnSearch’).

Editor de temas (I)

Descripción

Proporciona soporte para mantener múltiples apariencias para una aplicación, por ejemplo para diferentes divisiones corporativas.

Para ello se crean diferentes colecciones de objetos MIME para la aplicación BSP. Al especificar un tema para la aplicación, el servidor de aplicación seleccionará la colección de objetos adecuada.

Flujo del proceso

Para crear un tema:

1. Seleccionar la aplicación BSP
2. Pulsar el botón “Asignar tema” en la barra de botones de la aplicación.
3. Especificar el nombre del tema y grabar.
4. En el editor de temas, seleccionar la pestaña “Archivos” y pulsar el botón “Incluir Objetos”

Editor de temas (II)

Flujo del proceso (continuación)

5. Aparecerá un pop-up que muestra el repositorio MIME. Ahora basta con seleccionar aquellos objetos del repositorio que se desean y añadir el tema.
6. En el nuevo objeto añadido importar el nuevo gráfico.

Ahora podemos volver a la aplicación BSP y asignar el tema dentro de la pestaña de propiedades generales. Por defecto, la aplicación se ejecutará con este tema.

Objetos y programación OO

Objetos.

Un objeto es una sección de código que contiene datos y proporciona servicios. Los datos son los atributos del objeto. Y los servicios se conocen como métodos.

Normalmente los métodos actúan sobre datos privados (atributos o estado del objeto) que sólo son visibles a los métodos del objeto. De esta forma, los atributos no pueden cambiarse directamente por el usuario sino sólo a través de los métodos del objeto.

Clases

Las clases describen objetos. Desde un punto de vista técnico, los objetos son instancias en tiempo de ejecución de una clase. En teoría, se pueden crear cualquier número de objetos basados en una única clase. Cada instancia (objeto) tiene una identidad única y tiene su propio conjunto de valores para sus atributos.

Referencias a objetos

En un programa, una referencia a un objeto permite acceder a los atributos y métodos de un objeto.

(**Manual** : ABAP orientado a objetos – Manuel Fernández Arguelles AG)

Estructura de una clase (I)

Las clases son plantillas para crear un objeto, esto es una descripción abstracta de un objeto. Los atributos de los objetos se definen por los componentes de la clase, que definen el estado y el comportamiento de los objetos.

Una clase tiene componentes:

- Atributos
- Métodos
- Eventos
- Tipos

Constantes

Cada componenete tiene asignado una visibilidad.

- Public
- Protected
- Private

Estructura de una clase (II)

Atributos

Los atributos son campos de datos internos a la clase. El estado de un objeto se determina por el contenido de sus atributos.

Tipos de Atributos

Los atributos pueden ser:

- Atributos de instancia. El contenido de los atributos es específico de cada objeto.
- Atributos Estáticos. El contenido define un estado de la clase que es válido para todas las instancias. Sólo existen un conjunto de atributos estáticos por clase, siendo accesibles por todos los objetos instanciados de la clase.

Estructura de una clase (III)

Métodos

Los métodos son los internos de la clase que definen el comportamiento de un objeto. Pueden acceder a todos los atributos de la clase, lo que les permite modificar su contenido. Los atributos privados de una clase solo pueden cambiarse por métodos de la misma clase.

Tipos de método

- Métodos de instancia. Pueden acceder a todos los atributos de una clase y pueden lanzar todos los eventos de la clase.
- Métodos estáticos. Sólo pueden acceder a los atributos estáticos y lanzar los eventos estáticos.
- Métodos especiales. Además de los métodos normales existen dos métodos especiales denominados **CONSTRUCTOR** y **CLASS_CONSTRUCTOR**, que se crean automáticamente cuando se crea un objeto (**CONSTRUCTOR**) o cuando se accede por primera vez a los componentes de la clase (**CLASS_CONSTRUCTOR**).

Estructura de una clase (III)

Eventos

los objetos o clases pueden utilizar eventos para disparar métodos gestores de eventos en otros objetos o clases. Con los eventos, los métodos gestores determinan a qué eventos quieren reaccionar.

Tipos

Dentro de una clase se pueden definir tipo de datos propios. Estos tipos no son específicos de cada instancia, existiendo una sola vez para todos los objetos de la clase.

Constantes

Las constantes son atributos estáticos específicos. Se fijan los valores al declararlas y no pueden volver a modificarse. Estos tipos no son específicos de cada instancia, existiendo una sola vez para todos los objetos de la clase.

Manejo de objetos (I)

Creación de objetos

Antes de crear un objeto de una clase, se necesita declarar una variable de referencia a esa clase.

```
data <ref> type ref to <class>.
```

Puede crearse un objeto utilizando la sentencia:

```
create object <ref>.
```

Creación de mas de una instancia de una clase

En un programa, se pueden crear cualquier número de objetos de la misma clase, ya que son completamente independientes los unos de los otros, teniendo sus propios atributos. Cada sentencia CREATE OBJECT genera un nuevo objeto cuya identidad está definida unívocamente por la referencia.

Manejo de objetos (II)

Utilización de los componentes de un objeto

Componentes de la instancia <ref>:

Atributo: <ref> -> <attr>

Método: CALL METHOD <ref>-><meth>

Componentes estáticos de la clase <claa>

Atributo estático: <claa>=><attr>

Método estático: CALL METHOD <ref>=><meth>

Los componentes estáticos de una clase son accesibles sin llegar a crear un objeto de referencia.

Dentro de una clase puede referencia ME para acceder a los componentes individuales:

Atributo en la misma clase: ME-><attr>

Método en la misma clase : CALL METHOD ME-><meth>

Ejemplo VI

Creación de una clase de aplicación

Descripción

Las clases de aplicación son útiles en dos casos:

- Situaciones en las que se desea separar la lógica de la aplicación Web. En este caso se está proporcionando una interface pero se está ocultando la implementación.
- Situaciones en las que se desea compartir la lógica de la aplicación entre diferentes aplicaciones, ya sean Web o ABAP

NOTA: Las clases de aplicación se asignan directamente a la aplicación por lo que no es necesario crear una instancia de ella para poder utilizarla.

NOTA: Es ahora cuando el análisis previo de la aplicación es más importante. Para ello no vendría de más repasarse todos los temas de programación OO

Ejemplo VI

Creación de una clase de aplicación

Procedimiento

Para crear una clase de aplicación se utiliza la transacción SE24, o bien desde la propia SE80.

1. Ejecutar la transacción SE24 (Class Builder).
2. Crear una clase de aplicación global del tipo ABAP habitual. La clase debe permitir la creación de instancias de tipo publico.
3. Grabar y activar

En este punto la clase ya estaría creada, sólo falta añadirle los métodos , atributos y excepciones...

Ejemplo VI

Creación de una clase de aplicación

The screenshot illustrates the process of creating a class in the SAP ABAP development environment. It is divided into three main areas:

- Area 1 (Left):** The 'Transport Organizer' is selected in the left-hand pane. Below it, the 'Clase / Interface' option is chosen in the 'Jerarquía de aplicaciones' (Application Hierarchy) pane.
- Area 2 (Right):** The 'Tag Browser' is selected in the right-hand pane. The class name 'ZWEB_GEN_SESION' is visible in the dropdown menu.
- Area 3 (Center):** A dialog box titled 'Crear la clase ZSGRGEN2_PRUEBAS' (Create class ZSGRGEN2_PRUEBAS) is open. It contains the following fields:
 - Clase:** ZSGRGEN2_PRUEBAS
 - Descripción:** Objeto de datos bancarios
 - Creac. instancia:** Público
 - Tipo de clase:** Clase ABAP habitual, Clase excepción, Clase persistente
 - Final
 - Sólo modeladoButtons for 'Grabar' (Save) and a red 'X' are at the bottom.

Additional details: A context menu is open over the class 'ZWEB_GEN_SESION' in the right pane, with 'Clase' highlighted in red. An arrow points from this menu to the 'Clase' field in the dialog box. A red circle '3' is placed near the 'Tipo de clase' section of the dialog.

Ejemplo VI

Creación de una clase de aplicación

Procedimiento

Para crear un atributo.

- 1.- Ejecutar la transacción SE24, seleccionar la clase global
- 2.- En la pestaña “Atributos “ añadir un nombre al atributo, especificar el tipo de dato y el tipo de acceso.
- 3.- Grabar y activar la clase

Crear atributo

Tipo objeto: ZSRGEN2_GET

Atributo: Language

Descripción: Atributo del idioma

Visibilidad: Public, Protected, Private

Atributo: Constante, Estático, Instancia

Tipificación: LIKE, TIPO, TYPE REF TO, Entrada tipo directa

Tipo: BAPITRVXXX-LANGU

Sólo lectura

Valor inicial: []

Modelado

Crear [X]

Ejemplo VI

Creación de una clase de aplicación

Procedimiento

Para crear un método:

1. Ejecutar la transacción SE24, seleccionar la clase global
2. En la pestaña “Metodos “ añadir un nombre al método (`get_flights`), especificar el tipo de acceso (privado, público) el tipo de método (estático o de instancia) y la descripción.
3. Dentro de la pestaña del método tenemos dos botones, uno que nos dirige a los parámetros del método y otro a sus excepciones. En nuestro caso señalaremos dos parámetros:

```
importing carrid type string  
changing flight table type flighttab
```

Y una excepción:

```
NO_DATA_FOUND
```

Ejemplo VI

Creación de una clase de aplicación

Crear el método

Clase: ZHR_BANK_DETAILS

Método: CHECK_BANCO

Descripción: Funciones de chequeo de datos de la cuenta bancaria

Propiedades | Parámetros | Excep.

Visibilidad

Públ.

Protegido

Privado

Método

Estadístico

Instancia

Abstracto

Final

Prog.trat.eventos p.

Clase/Interface: _____

Evento: _____

Modelado

Crear

Ejemplo VI

Creación de una clase de aplicación

Procedimiento (continuación)

- Implementar el código del método.

```
select *  
from sflight  
into corresponding fields of table flights up to 100 rows  
where carrid = carrid  
if sy-subrc ne 0.  
raise no_data_found.  
endif
```

5. Grabar y activar la clase.

Ejemplo VI

Creación de una clase de aplicación

Descripción

Antes de poder utilizar la clase de aplicación hay que introducirla en la aplicación BSP

Procedimiento

Para asignar una clase de aplicación a una aplicación BSP

1. Seleccionar la aplicación BSP en la transacción SE80
2. Seleccionar la pestaña “ Propiedades “
3. Introducir el nombre de la clase en el campo “ Clase de Aplicación “
4. Grabar y activar.

Ejemplo VI

Creación de una clase de aplicación

The screenshot displays the SAP Web Application Builder interface for modifying a BSP application. The main window title is "Web Application Builder: Modificar aplicación BSP". The application name is "zbsp_flight" and its status is "Activo/revisado".

The left sidebar contains several toolbars: "Repository MIME", "Browser de Repository", "Sistema Info Repository", "Tag Browser", and "Transport Organizer". Below these, there is a section for "Aplicación BSP" with a dropdown menu showing "zbsp_flight".

The main configuration area is divided into two tabs: "Atributos" and "Navegación". The "Atributos" tab is active, showing the following fields:

Descripción breve	Prueba		
Creado por	SRGEN2	Fecha creación	02.09.2004
Modificado por	SRGEN2	Fecha modificación	02.09.2004
Paquete	\$TMP		
Idioma maestro	ES		
Nombre interno	ZBSP_FLIGHT		
Imagen inicial BSP	search.htm		
Clase de aplicación	ZWEB_LIQ_SESION		
Tema			

Below the fields, there are two checkboxes: "Con estado" (checked) and "Soporta integración portal" (unchecked).

The bottom status bar shows the user "SRGEN2", the session "espux008", and the instance "INS".

Ejemplo VI

Creación de una clase de aplicación

Utilizamos ahora el método de la clase que hemos creado en la inicialización de los datos:

1. Seleccionar la aplicación BSP en la transacción SE80
2. Seleccionar el evento OnInitialization de la página display.htm
3. Cambiar el código del evento por lo siguiente:
 Call method application ->get_flight
 exporting carrid = carrid
 Changing flights = flights
 Exceptions no_data_found = 1.
4. Falta capturar las excepciones, navegando a una página de error.

Aplicación BSP con estado - Stateful

Una aplicación BSP se ejecuta en el único contexto (roll area) como cualquier transacción SAP. Cualquiera de los datos introducidos por el usuario o determinados por la aplicación son accesibles en el contexto de la sesión.

Todas las peticiones HTTP independientes se agrupan en tiempo de Ejecución utilizando *cookies de sesión*.

Ventajas:

- Programación simplificada. No hay que volver a recalcular los datos, lo que lleva a una mejora sustancial del rendimiento.

Desventajas :

- Hay que compensar el factor tiempo de ejecución con el factor memoria. El número de sesiones en paralelo se restringe por la cantidad de memoria disponible. Cuando ésta se agota, se rechazará cualquier acceso de nuevos usuarios hasta que haya recursos disponibles.
- Los navegadores no realizan la desconexión del sistema y por tanto una sesión permanecerá abierta hasta que sea liberada por el mecanismo de timeout.

Aplicación BSP sin estado – Stateless (I)

Descripción:

Las aplicaciones BSP que no mantiene el estado sólo consumen recursos del WAS en el momento en que la petición está siendo procesada... Cuando se ha terminado, todos los recursos se devuelven al sistema para un uso futuro en otras peticiones.

Ventajas:

Permite un escalamiento óptimo de las aplicaciones.

Desventajas:

Liberar el contexto de la aplicación implica que es posible que haya que obtener los mismos datos varias veces con la consiguiente repercusión en el tiempo de respuesta.

Aplicación BSP sin estado – Stateless (I I)

Descripción

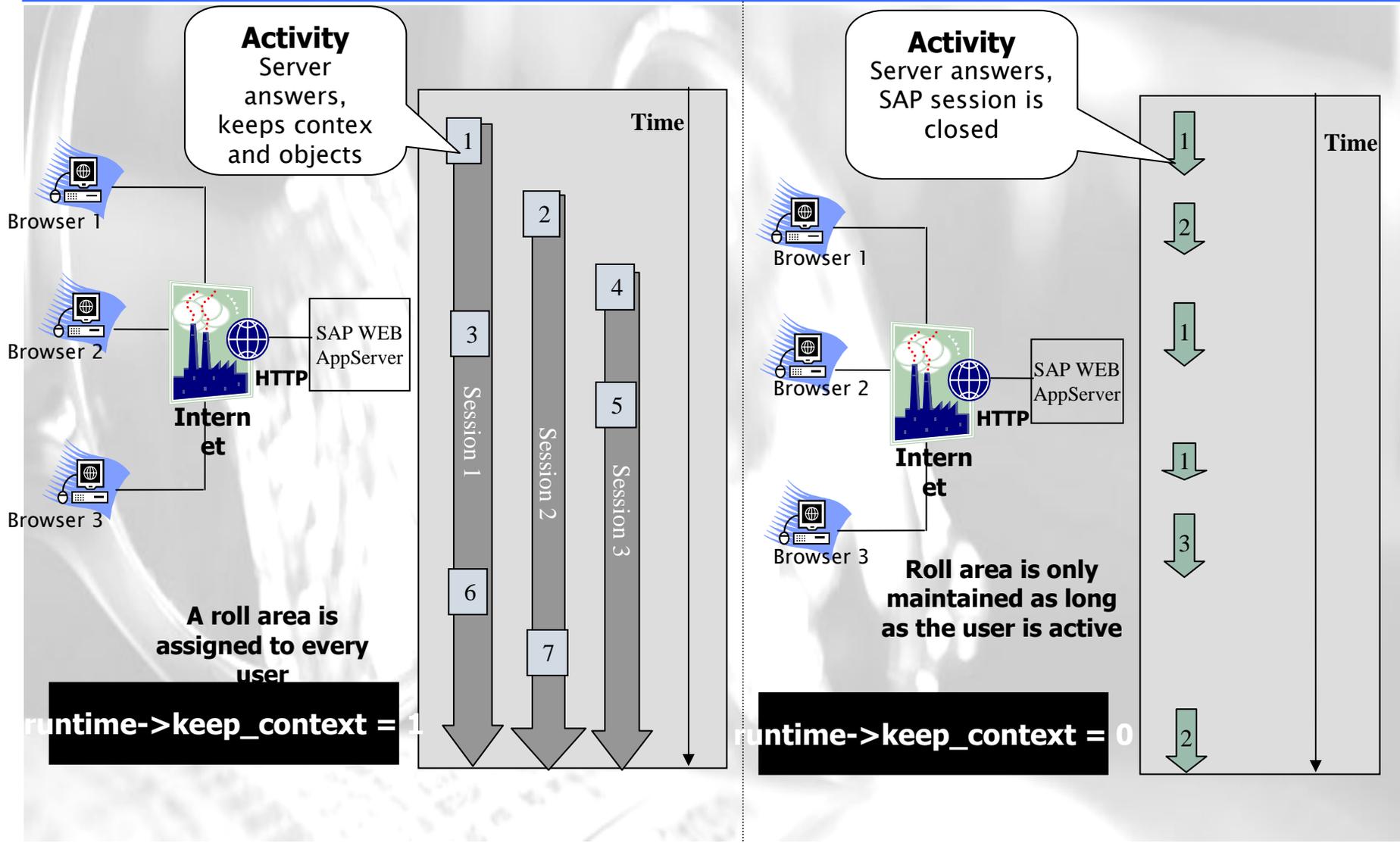
Para mantener información de una aplicación BSP sin estado, deberá programarse la funcionalidad para conservar y devolver los datos entre llamadas

Las soluciones técnicas son:

- Almacenamiento temporal en el Website con campos HTML ocultos:
`<input type=hidden name=“customerID” value=“4711”>`
- Cookies en el lado cliente
- Cookies en el lado servidor

Almacenamiento temporal en tablas específicas de la aplicación.

Stateful vs. Stateless (I)



Stateful vs. Stateless (II)

Manteniendo el estado

El servidor mantiene la sesión.

Penalización en memoria .Limita el número de sesiones concurrentes.

Simplifica el desarrollo.

Pensando para optimizar el acceso a Datos. Sólo se obtiene una vez.

No manteniendo el estado

Después de cada petición el servidor se “ olvida “ de todo.

Penalización en factor tiempo. Permite un número mas alto de usuarios concurrentes.

Debe codificarse métodos para guardar la información entre peticiones : campos ocultos, cookies, etc...

Como hacer una BSP Stateful o Stateless

Puede hacerse durante la fase de desarrollo o bien en tiempo de ejecución.

- **Durante el desarrollo.** En la transacción SE80, seleccionar la aplicación BSP. Dentro de las propiedades de la aplicación, existe un checkbox que indica si la aplicación debe mantener estado o no.
- **En tiempo de ejecución.** Se hace mediante la interface de programación IF_BSP_RUNTIME. En cualquier momento puede cambiarse de uno a otro modo fijando o borrando el atributo keep_context, o que sobrescribirá el valor que se le hubiera asignado durante la fase de desarrollo.

Nota: En la práctica, en ninguna aplicación tiene un modelo único de trabajo en todas las situaciones posibles. Un caso típico es una tienda on line, mientras que durante la fase de navegación por el catálogo no tiene sentido mantener el estado, durante la fase de pedido, se mantiene muy a menudo el estado.

Cookies de cliente

Descripción.

Una cookie es un mecanismo HTTP que permite a un servidor enviar Cantidades limitadas de información a un cliente HTTP. El cliente envía De retorno la cookie al servidor con la información. Se utiliza entonces Para mantener el estado de la conversación cliente-servidor en aplicaciones que no conservan el estado.

Consideraciones

En el manejo de cookies en el lado cliente debe considerarse:

- La seguridad. Al estar almacenadas como ficheros de texto en el disco duro, son fácilmente alterables. Hay que verificar siempre la información que viene en una cookie.
- El cliente puede tener desactivada la opción de trabajar con cookies.
- La información que puede almacenarse es limitada:
 - 4 kilobytes por cookie.
 - 300 cookies en total.
 - 20 cookies por servidor o dominio.

Cookies de servidor (I)

Definición

Los cookies de servidor son datos persistentes similares a las cookies del lado cliente que se almacenan en la BBDD. Los cookies de servidor no tienen limitaciones de tamaño y cantidad que tienen los del lado del cliente.

Por razones técnicas, cada cookie solo se puede almacenar de una de las siguientes formas:

- * Como campo
- * Como estructura
- * Como tabla interna

La clase `CL_BSP_SERVER_SIDE_COOKIE` proporciona los métodos para crear, obtener, borrar y gestionar las cookies del servidor.

Nota: Al recobrar una cookie, debe hacerse sobre la misma estructura sobre la que se almacenó, en caso contrario se provocará un error.

Cookies de servidor (II)

Para crear una cookie de servidor:

Call method `cl_bsp_server_side_cookie=>set_server_cookie` exporting

Name	=“FLIGHTS”
Application_namespace	=runtime->application_namespace
Applicarion_name	=runtime->application_name
Username	=usuario
Session_id	=runtime->session_id
Expiry_date_abs	=exp_date
Expiry_time_abs	=sy-uzeit
Data_name	=“FLIGHTS”
Data_value	=flights

Cookies de servidor (III)

Para obtener el valor de una cookie de servidor:

Call method `cl_bsp_server_side_cookie=<get_server_cookie`

`exporting`

`Name`

`=“FLIGHTS”`

`Application_namespace`

`=runtime->application_namespace`

`Application_name`

`= runtime->application_name`

`Username`

`=usuario`

`Session_id`

`=runtime->session_id`

`Data_name`

`=“FLIGHTS”`

`Importing`

`Expiry_date`

`=exp_date`

`Expiry_time`

`=exp_time`

`Changing`

`Data_value`

`=flights.`

Cookies de servidor (IV)

Para borrar una cookie de un servidor:

Call method `cl_bsp_server_side_cookie=<delete_server_cookie`
`exporting`

<code>Name</code>	<code>=“FLIGHTS”</code>
<code>Application_namespace</code>	<code>=runtime->application_namespace</code>
<code>Application_name</code>	<code>= runtime->application_name</code>
<code>Username</code>	<code>=usuario</code>
<code>Session_id</code>	<code>=runtime->session_id</code>

Ejemplo VII

Creación de una cookie de cliente

Procedimiento

Para crear una cookie de cliente desde la propia SE80:

1. Crear una aplicación BSP, Z_RT405_XX_COOKIE
2. Crear dos páginas, una llamada default.htm y otra display.htm.
3. El código de la página default.htm contiene un campo de introducción de datos y un botón de aceptación, que dispara el evento `OnInputProcessing` que es el encargado de almacenar la cookie persistente en el lado cliente y navegar a la página *display.htm*.
4. En la página display.htm obtenemos primero la cookie almacenada en el lado cliente, en el evento *OnInitialize*, almacenándola en la variable `carrid`. Posteriormente la mostramos en la página.

Ejemplo VII

Creación de una cookie de cliente

El código de HTML de la página default.htm queda como sigue:

```
<%@page language="abap"%>
<html>
  <head>
    <link rel="stylesheet"
      href="../../sap/public/bc/bsp/styles/sapbsp.css">
    <title> Página por defecto </title>
  </head>
  <body>
    <form>
      Compañía: <input name="CARRID"/>
      <input type="submit" name="OnInputProcessing" value="Cookie...>
    </form>
  </body>
</html>
```

Ejemplo VII

Creación de una cookie de cliente

1. En el evento OnInputProcessing de la página default.htm definimos las siguientes variables:

```
data:      carrid      type string,  
          expira      type string,  
          ts          type bsptimestamp,  
          ttss(14)    type c,
```

2. Obtenemos el dato del formulario y fijamos la fecha de caducidad de la cookie:

```
ts-date = sy-datum + 1  
ts-time = sy-uzeit.  
ttss    = ts  
carrid  = request->get_form_field( "carrid" )
```

Ejemplo VII

Creación de una cookie de cliente

3. Cambiamos el formato de fecha a un formato de fecha HTTP utilizando la clase CL_BSP_UTILITY:

```
class cl_bsp_utility definition load  
  expira = cl_bsp_utility=>date_to_string_http ( timestamp = ttss ).
```

4. Escribimos la cookie de cliente y navegamos a la página siguiente
call method runtime->server->response->set_cookie

```
exporting name      =      "carrid"  
  path              =      ' / '  
  value             =      carrid  
  expires           =      expira  
  secure            =      ' 0 '  
  navigation->next_page ('DISPLAY')
```

Ejemplo VII

Creación de una cookie de cliente

El código de HTML de la página default.htm queda como sigue:

```
<%@page language="abap"%>
<html>
  <head>
    <link rel="stylesheet"
      href="../../sap/public/bc/bsp/styles/sapbsp.css">
    <title> listado de vuelos </title>
  </head>
  <body>
    <p> La cookie vale <%=carrid %> </p>
  </body>
</html>
```

Ejemplo VII

Creación de una cookie de cliente

1. En el evento OnInitialization de la página display.htm introducimos el siguiente código:

Call method c->server->request->get_cookie

Exporting name = 'carrid'

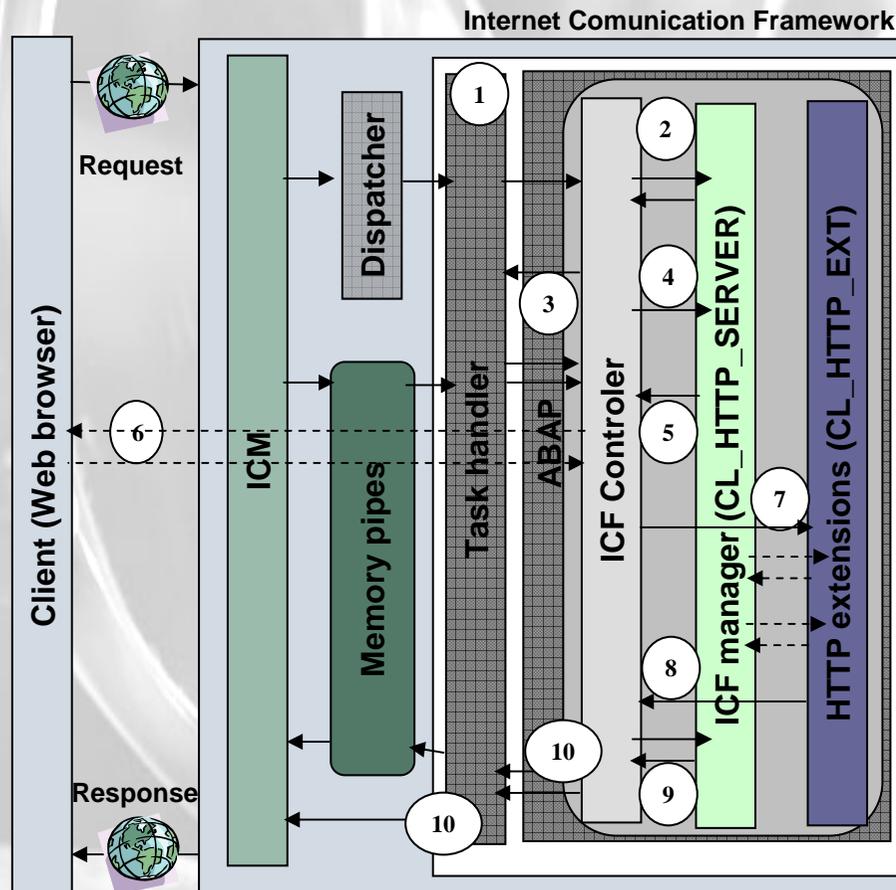
Importing value = carrid.

Anexo I.-Arquitectura del WAS

En el Anexo SAP Web AS se da una visión más detallada de la arquitectura del Web Application Server

Modelo de implementación _ Servidor

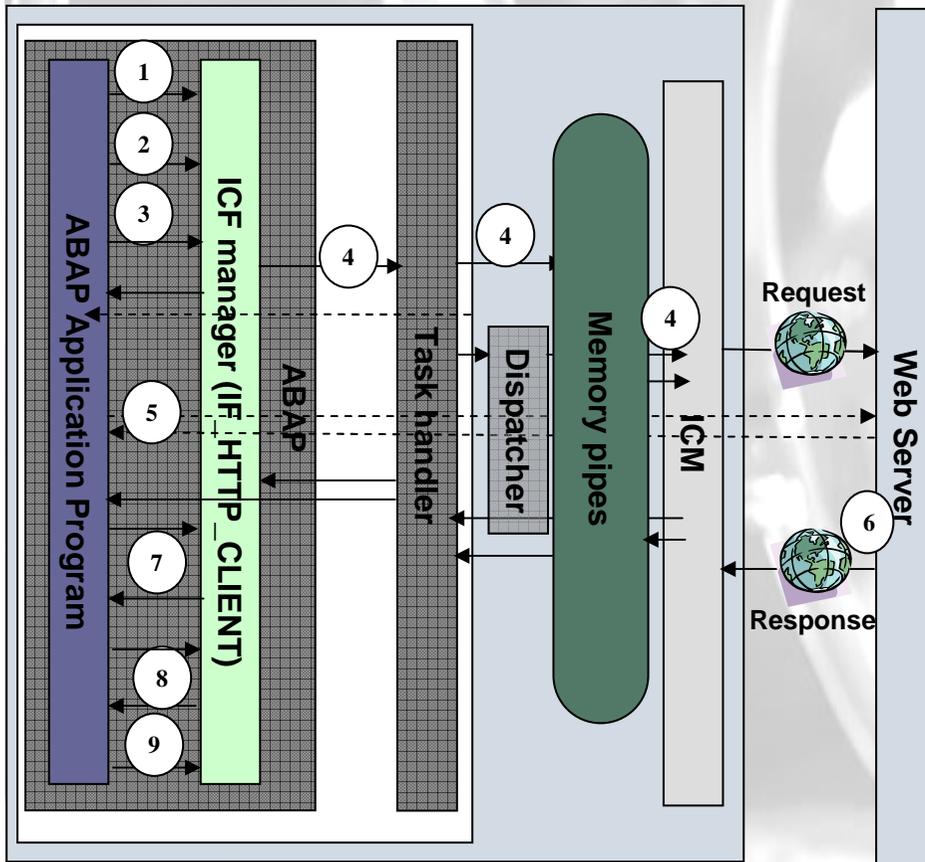
HTTP client HTTP Server (SAP Web Application Server)



1. Llamada al módulo de Función HTTP_DISPATCH-REQUEST.
2. Se crea un objeto de la clase CL_HTTP_SERVER(server control block).
3. Se lee la petición HTTP.
4. Se rellena el Server Control Clock con los datos de la petición HTTP.
5. Se selecciona el gestor de la petición en función de la URL (SICF).
6. El cliente se conecta.
7. Se llama al gestor de la petición HTTP, que gestiona los datos, llama las aplicaciones...
8. El ICF abandona el control en este punto, pudiendo dedicarse a otras tareas.
9. Se crea otra cadena HTTP (serialización de la respuesta)
10. El resultado se escribe en los memory pipes y se devuelve al cliente via ICM

Modelo de implementación _ Cliente

HTTP client



HTTP Server

1. Se crea un objeto de la clase CL_HTTP_Client (client control block)
2. Se rellena el Client Control block con los datos de la petición y se envía. Se abre la conexión y la petición se convierte en un HTTP data stream
3. La petición se envía al servidor vía ICM
4. Si el servidor HTTP requiere autenticación, el cliente ejecuta el logon
5. El servidor HTTP genera una respuesta y la devuelve.
6. Se llama al método RECEIVE, rellenando los datos del client Control Block.
7. Se procesan los datos encontrados en el atributo RESPONSE.
8. Se cierra la conexión con el método CLOSE

Parámetros URL específicos del sistema

Parámetro URL	Valor	Descripción
Sap-sessioncmd	open	Reinicia la aplicación pasada en la URL
	close	Finaliza la aplicación y dirige al navegador a la dirección definida en sap-sexiturl
Sap-sexiturl	*	Especifica la dirección a la que se dirige el navegador cuando la aplicación finaliza
Sap-theme	*	Reemplaza el tema de una aplicación BSP
Sap-client	*	Especifica al cliente al que se le efectúa la conexión
Sap-user	*	Especifica el usuario que realiza la conexión
Sap-password	*	Especifica la contraseña para la conexión
Sap-language	*	Especifica el lenguaje de conexión

Anexo II – Llamadas remotas



Creación de un destino RFC

Flujo del proceso

Para crear un destino RFC.

1. Crear un usuario genérico de tipo CPIC con la transacción SU01 en la máquina destino
2. En el servidor WAS, con la transacción SM59, pulsar el botón crear.
3. Introducir el nombre, la descripción y seleccionar como tipo Conexión R/3 (3)
4. Grabar e introducir la dirección IP de la máquina de destino en la Opciones Técnicas de la conexión
5. En las opciones de seguridad, introducir en los datos de entrada al Sistema, los datos del usuario genérico creado en el paso 1
6. Grabar y probar la conexión

Ejecución de funciones remota

Flujo del Proceso

Para ejecutar funciones remotamente:

1. Crear un destino RFC en el WAS.
2. Si fuera necesario, crear los correspondientes módulos de función en el sistema, habilitándolos para su ejecución en remoto.
3. En ABAP, se puede especificar el sistema desde el que queremos obtener los datos al llamar a un Módulo de Función o una BAPI modificando la llamada:

call function <NOMBRE DEL MF>

destination <DESTINO>

...

Donde <DESTINO> es el nombre del destino RFC que hemos creado en el primer paso.

Comprobaciones de Autorización RFC

1. Para ejecutar un módulo de función RFC, el usuario de conexión remoto necesita las autorizaciones correspondientes para el objeto de autorización S_RFC
2. El objeto de autorización S_RFC permite definir los grupos de funciones para los cuales el usuario de conexión RFC tiene permitido ejecutar RFC. El usuario sólo podrá ejecutar módulos de función que se encuentren dentro de estos grupos de función.
3. El objeto de autorización S_RFC contiene tres campos:

CAMPO	VALOR	DESCRIPCION
RFCNAME	*	Nombre del grupo de funciones
ACTVT	16	Actividad
RFC_TYPE	FUGR	Tipo de objeto RFC

BAPIs (I): Definición

Business Application Programming Interface (BAPI) es un interfaz que proporciona acceso a datos y a procesos de negocio en sistemas R/3.

Las BAPIs son definidas como métodos del API de SAP como business object. Estos tipos de objeto de negocio y sus BAPIs son descritas y almacenadas en el Business Object Repository (BOR).

Una BAPI se implementa como un módulo de función, que está definido y almacenado en el Function Builder.

En definitiva, una BAPI es un módulo de función (subconjunto de RFC) que se utiliza para llamar a programas externos y para acceder a datos.

BAPIs (II): Utilización en WAS

Las BAPIs se utilizan en WAS para acceder a los datos de la misma forma que se pueden utilizar select a tablas de BBDD, RFC a otros sistemas , etc. Es una forma más de manipular los datos sólo que ofrece la ventaja de que realizan una serie de comprobaciones acerca de permisos y accesos de los usuarios a ciertos datos (ej.- módulo HR).

Esto es una ventaja puesto que de esta forma a la hora de programar BPS nos ahorra trabajo (validaciones de campos, errores del sistema, etc.)

La forma de utilización en WAS es encapsularlas en un método dentro de un objeto. De esta forma es accesible por cualquier BSP.

BAPIs (III): Creación del Objeto

The screenshot shows the SAP ABAP Class Editor interface. The title bar reads "Generador cl. Clase ZHR_SS_DETAILS Modif.". The left sidebar shows the class structure for "ZHR_SS_DETAILS", with the "OBTENER_AFILIACION" method selected. The main editor displays the following ABAP code:

```

Método          OBTENER_AFILIACION          Activo(tratado)
METHOD obtener_afiliacion .
*-----*
* El método obtiene los datos de afiliación a la Seguridad Social de *
* un usuario determinado mediante su número de empleado y la fecha del *
* sistema. *
* Creación: SRGEN2 (NETCHECK S.A)                               Fecha: 21.07.2004 *
*-----*

CALL FUNCTION 'Z_BAPI_AFILIACION_SS' DESTINATION 'NONE'
  EXPORTING
    employeenummer = employeenummer
    timeintervallow = timeintervallow
    language        = language
  IMPORTING
    werks           = werks           "centro trabajo
    gesch          = gesch          "sexo
    gbdat          = gbdat          "fecha nacimto
    gbort          = gbort          "lugar nacimto
    natss         = natss         "numero ss
    antiguedad    = antiguedad    "fecha alta
    berkt         = berkt         "categoria profesional
    grcot         = grcot         "grupo tarifa
  EXCEPTIONS
    OTHERS        = 1.

ENDMETHOD.
  
```

The status bar at the bottom indicates "Línea 10 columna 1" and "Línea 1 - línea 29 de 29 líneas". The user ID "SRGEN2" and system "espx008" are visible in the bottom right corner.

BAPIs (IV): Acceso desde el BSP

The screenshot displays the SAP IDE interface for developing a BSP page. On the left, the 'Objeto' (Object) browser shows the project structure: 'zverdata' containing 'Páginas con lógica proceso' (Pages with process logic), which includes the selected file 'afiliacion.htm'. The main editor window shows the HTML and ABAP code for this page.

Page Information: Página: afiliacion.htm, Estado: Activo(Revisado)

Code Content:

```

</tr>
</table>
</form>
</div>
<htmlb:form id="formu"><center>
<%
**** CARGA LOS DATOS DE LA AFILIACION SS *****
DATA: centro      TYPE BAPIP0001-SNAME,
      sexo        TYPE BAPIP0001-KOSTL,
      fecha_nac   TYPE BAPIP0002-GBDAT,
      lugar_nac   TYPE BAPIP0002-GBORT,
      numeross    TYPE BAPIP0001-VDSK1,
      fecha_alta  TYPE BAPIP0002-GBDAT,
      categoria   TYPE BAPIP0002-NAME2,
      tarifa      TYPE BAPIP0002-KNZNM.

data: objafiliacion type ref to ZHR_SS_DETAILS.

CREATE OBJECT objafiliacion.
objafiliacion->EMPLOYEEENUNBER = application->number.
objafiliacion->LANGUAGE         = application->language_int.

CALL METHOD objafiliacion->OBTENER_AFILIACION
  EXPORTING
    TIMEINTERVALLOW = sy-datum
  IMPORTING
    WERKS      = centro      "centro trabajo
    GESCH      = sexo       "sexo
    GBDAT      = fecha_nac   "fecha nacimto
    GBORT      = lugar_nac   "lugar nacimto
    NATSS      = numeross    "numero ss
    ANTIGUEDAD = fecha_alta  "fecha alta
    BERKT      = categoria   "categoria profesional
    GRCOT      = tarifa      "grupo tarifa

*****
%>

```

Status Bar: Línea 19 columna 1, Línea 14 - línea 50 de 158 líneas

BAPIs (VI): Resultado Final

Cuando se ejecuta el BSP desde el navegador, el resultado que se obtiene es el siguiente:

Afiliación Seguridad Social

| Español | English |

Datos Personales

NIP: 00013283 **Nombre:** Monroe Curie, Luisin

Datos Afiliación

Centro de trabajo	Madrid - Tres Cantos
Sexo	VARÓN
Fecha de nacimiento	07.11.1964
Lugar de nacimiento	Madrid
Nº de afiliación de la SS	280369328605
Fecha de antigüedad	01.10.1988
Categoría Profesional	Técnico de Administración Esp.
Grupo de tarifa	04

Nota

En caso de existir algún dato incorrecto por favor ponte en contacto con el responsable de tu nómina. Si deseas hacerlo en este momento por e-mail puedes hacerlo pinchando **"aquí"**

[Subir](#) 

Anexo III – Servicios DDIC



Servicios DDIC para aplicaciones BSP (I)

Descripción

Servicios que se utilizan para proporcionar a las aplicaciones Web información y definiciones ya existentes en el sistema. Son servicios de diccionario que se basan en las propiedades del tipo de un objeto de datos.



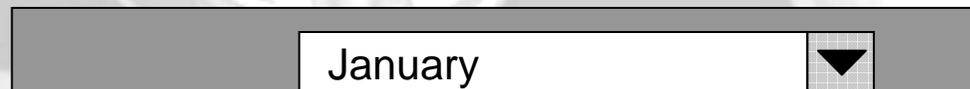
A screenshot of a web form. On the left, the label 'Airline' is positioned above a horizontal line. To the right of this line is an empty rectangular input field. Below the input field, the text 'Airline Code' is displayed.

Ayuda de campo



A screenshot of a web form. On the left, the label 'Airline' is positioned above a horizontal line. To the right of this line is an input field containing the letter 'a'. Below the input field, a dropdown menu is open, displaying a list of airline names: 'Air France', 'Alitalia', and 'American Airlines'.

Histórico de valores



A screenshot of a web form. It shows a dropdown menu with the text 'January' selected. To the right of the text is a small square button containing a downward-pointing arrow.

Valores posibles

Servicios DDIC para aplicaciones BSP (II)

El objeto runtime permite acceder a la información relevante en tiempo de ejecución. Esto incluye la información necesaria para la visualización, definida en el servicio `ddic_utils`.

Los servicios fundamentales que `ddic_utils` nos proporciona son :

GET_FIELD_LABEL

Devuelve la etiqueta almacenada en el diccionario para el tipo de dato.

GET_QUICKINFO

Devuelve la etiqueta mas corta almacenada en el diccionario para el tipo de dato.

GET_HISTORY_ID

Muestra la historia global de los últimos valores introducidos en el tipo de dato.

GET_LOCAL_HISTORY_ID

Muestra la historia de los últimos valores introducidos en el tipo de dato para la aplicación

GET_SIMPLE_HELP_VALUES

Tabla de valores posibles de selección

Ejemplo IX

Ayuda de campo

Descripción

Vamos a introducir una ayuda dentro del campo carrid de la pantalla de búsqueda de la aplicación de ejemplo.

Procedimiento

Los valores de la ayuda los vamos a almacenar en dos nuevos atributos de la página: carrid_title y carrid_page, que serán de tipo string

En HTNL sólo hay que cambiar la sentencia:

```
<span title="<%=carrid_title%>"><%=carrid_label%></span>&nbsp;   <input type=text name= "carrid" value="carrid_label">
```

para que tome los valores de ayuda que vamos a obtener en el evento de inicialización de los datos de la pantalla.

Ejemplo VII

Ayuda de campo

Rellenemos la etiqueta del campo en el evento OnInitialization:

try.

```
carrid_title =
```

```
runtime->ddic_utils->get_field_label ( data_object_ref = dataref  
                                        langu =sy-langu)
```

Catch cx_bsp_services into l_ex_services

```
texto_error = l_ex_services->get_text ().
```

```
call method page-> messages->add_message
```

```
exporting condition = `CARRID_TITLE`
```

```
    message = texto_error
```

```
    severity = CL_BSP_MESSAGES=>CO_SEVERITY_ERROR
```

```
Carrid_label = 'CARRID_TITLE'.
```

```
endtry
```

Ejemplo IX

Histórico de valores

Descripción

Vamos a introducir una lista de las últimas selecciones del usuario del campo carrid de la pantalla de búsqueda de la aplicación de ejemplo. Esta función sólo puede utilizarse si la permite el navegador (IE5.0 o superior)

Procedimiento

Los valores de la ayuda los vamos a almacenar en un nuevo atributo de la página. carrid_local_his

En elHTML sólo hay que cambiar la sentencia:

```
<input type = "text"  
      name = "carrid "  
      vcard_name="<%=carrid_local_his%> " value="" />
```

Para que tome los valores que vamos a obtener en el evento de Inicialización de los datos de la pantalla.

Ejemplo X

Histórico de valores

Rellenemos la historia en el evento OnInitialization:

Data: dataref type ref to data,

texto_error type string

l_ex_services type ref to cx_bsp_services.

Get reference of flight-carrid into dataref.

try.

carrid_local_his =

runtime->ddic_utils->get_local_history_id (data_object_ref = dataref

Catch cx_bsp_services into l_ex_services

texto_error = l_ex_services->get_text ().

call method page-> messages->add_message

exporting condition = 'HISTORY_ID'

message = texto_error

severity = CL_BSP_MESSAGES=>CO_SEVERITY_ERROR

carrid_local_his = 'sap.myapplication.carrid'

endtry.

Ejemplo XI

Valores posibles para un campo

Descripción

Vamos a utilizar el método **GET_SIMPLE_HELPVALUES** para definir una lista de valores posibles de selección de una compañía aérea en la pantalla de búsqueda.

Procedimiento

Los valores los vamos a almacenar en un nuevo atributo de la página : **airlinest** de tipo **SHSVALTAB**.

Ejemplo XI

Valores posibles para un campo

En el HTML de la página de búsqueda modificamos un poco la pantalla search.htm,

```
<% if message ->num_messages ( ) = 0  
    data: wa type SHSVALSTR. %>
```

Línea aérea:

```
< select name = " AIRLINE ">  
<% Loop at airlinelist into wa. %>  
    <option value=" <%=wa-key%> "> <%=wa-value%> </option>  
<% Endloop. %>  
<% else. %>  
No existen líneas aéreas disponibles.  
<% endif. %>
```

Ejemplo XI

Valores posibles para un campo

Rellenamos la lista en el evento OnInitialization:

```
Data: dataref type ref to data,  
      texto_error type string  
      l_ex_services type ref to cx_bsp_services.
```

Get reference of flight-carrid into dataref.

try.

```
Call method runtime-ddic_utils->get_simple_helpvalue  
exporting data_object_ref = dataref  
changing helpvalue_tab    = airlinelist
```

Catch cx_bsp_services into l_ex_services

```
texto_error = l_ex_services->get_text ().
```

```
call method page-> messages->add_message
```

```
exporting condition = 'AIRLINES'
```

```
message = texto_error
```

```
severity = CL_BSP_MESSAGES=>CO_SEVERITY_ERROR
```

Anexo IV – Conexiones anónimas



Conexión anónima a un servicio

Flujo del proceso

Para conectarse anónimamente a un servidor:

1. Crear un usuario de servicio genérico, webguess, con la transacción SU01
2. En la transacción SICF, crear un servicio llamado bc/bsp/sap/<App BSP> con las siguientes características:
 - En el nivel del nodo bsp, poner como gestor de eventos CL_HTTP_EXT_BSP.
 - En el nivel del servicio <App BSP> introducir en los datos del servicio los datos de conexión del usuario genérico creado en la transacción SU01



Anexo V – Java Script



Ejemplo XIII

Scripts de servidor en JavaScript

Descripción

Vamos a mostrar la utilización de JavaScript para desarrollo de scripts en el lado del servidor

Procedimiento

Desde la transacción SE80

1. Crear una nueva página llamada display_js.htm como copia de la página display.htm
2. Seleccionar la página display_js.htm y cambiar la directiva BSP que fija el tipo de lenguaje de script a javascript.
3. Modificar el bucle que muestra los datos para que utilice js.

Ejemplo XIII

Scripts de servidor en JavaScript

JavaScript

```
<tabla>
<%for (i=0; i<flights.length; i++) %>
  <tr>
    <td><%= flights[i].carrid %></td>
    <td><%= flights[i].connid %></td>
    <td><%= flights[i].fldate %></td>
  </tr>
<% } %>
</tabla>
```

ABAP

```
<tabla>
<%Loop al fligths into flight %>
  <tr>
    <td><%= flight-carrid %></td>
    <td><%= flight-connid %></td>
    <td><%= flight-fldate %></td>
  </tr>
<% endloop %>
</tabla>
```

AYUDASS!!!

Links Javascript:

<http://www.forsdelweb.com/showthread.php?postid=261837#post261837>

<http://freak.jorgeferrer.com/conquistalaweb/dinamicos/javascript/efectos.html>

<http://www.mundojavascript.com/ventana/>



Foros del Web > Tecnologías del Web
FAQs JavaScript

Responder

02-12-2002, 06:38 PM

Kaopectate
Moderador

30.- Cerrar una ventana principal

P: ¿Como puedo cerrar una ventana de confirmación?

Créditos: Carlitos

Nota: Una versión mas sencilla y [115](#).

R: [\[ver ejemplo\]](#)
El siguiente script se basa en un código de otras plataformas.

```
Código PHP:  
<html>  
<!-- Aportado por: Carlitos -->  
<head>  
<script language="JavaScript">  
function salir() {
```

Alertas	Animaciones	Barra de estado	Calculadoras	Cookies
Novedades	Mundo Javascript		1242 días al servicio de los webmasters hispanos	Efectos
> > TOP 10 << <<			51 usuarios on-line	Formularios
Ventana	Relojes/calendarios	Menús	Marquesinas/Scroll	Imágenes
Utilidades	Herramientas Webmaster	Participa en el Ranking de webs	Recibe por e-mail nuestras actualizaciones con enlaces directos a los nuevos scripts.	Juegos
Directorio recursos y tutoriales GRATIS para webmasters			<input type="text"/> <input type="button" value="Suscribirse"/>	DESCUBRE LAS 6 VENTAJAS HOME ENGLISH
Portátiles		Especialistas en Ordenadores Portátiles Desde 685 € + IVA. Envío GRATIS Hasta tres años de garantía on-site		3 RAZONES para confiar tu hosting a arsys.es Servicios para tu web ATENCIÓN.- El servicio de intercambio de banners será suspendido indefinidamente el día 1/10/2003 Contenidos para tu web
PRECIOS10.COM	Ordenadores Portátiles a precio de fábrica con envío incluido	Ventana MOVIENDO LA VENTANA Al abrirse la ventana esta se desplaza de izquierda a derecha y de arriba a abajo, llamando la atención del visitante. VENTANA QUE SE ENCOGE Cuando se hace click sobre el botón o link la ventana se encoge y se vuelve a abrir.		
Precios10.net	Accesorios para portátiles Almacenamiento y Backup's Cámaras digitales Videocámaras digitales Componentes Consola XBOX nuevo	Portátiles Benq en Precios10.com, y como siempre a precios de		