



Departamento de Lenguajes y
Sistemas Informáticos



Universitat d'Alacant
Universidad de Alicante

Características de las tecnologías de web scripting

Programación en Internet
Curso 2007-2008

Programación en Internet – Curso 2007-2008

Índice

- Características generales
- Características web
- ASP, CFM, JSP y PHP
- Comparativa
- Estadísticas de uso

2

Características generales (I)

- Características “exigibles”
- Portabilidad: distintas plataformas (sistema operativo, servidor web, sistema gestor de bases de datos, etc.)
- Lenguaje de programación interpretado: 99% de los casos no se necesita mucha velocidad
- Permita usar código compilado para el 1% restante
- Conectividad con bases de datos:
 - Distintos sistemas
 - Independencia del sistema

3

Características generales (y II)

- Capacidad de reutilización de código: creación de componentes (librerías o módulos)
- Acceso al sistema de ficheros del servidor
- Gestión de errores (tratamiento de excepciones): capturar un error, tratarlo y mostrar un mensaje de error adecuado

4

Características web (I)

- Gestión de sesiones: persistencia de información individual de un usuario entre distintas peticiones (páginas)
- Acceso a las variables de:
 - Entorno
 - CGI (servidor, cliente y petición)
- Procesamiento de la entrada de datos: descodificación parámetros (tanto POST como GET)

5

Características web (y II)

- Soporte funcionalidades de Internet:
 - Envío/recepción de correo electrónico
 - Envío/recepción de ficheros mediante FTP
 - Carga/descarga de ficheros
 - Seguridad: encriptación, gestión de usuarios y contraseñas
- HTML + script entremezclado: la página web contiene código (CGI: código que genera una página web) → Si se estructura correctamente, se puede modificar el diseño sin alterar la lógica

6

ASP, CFM, JSP y PHP (I)

- Diferentes tecnologías con un mismo propósito: proporcionar contenido dinámico en la web
- Todas se basan en:
 - Un *script* se ejecuta en el servidor web
 - Puede recibir información del usuario
 - Produce una página web dinámica
 - El *script* se encuentra intercalado con el código de una página HTML

7

ASP, CFM, JSP y PHP (II)

- Emplean componentes o funciones para añadir más funcionalidades:
 - Acceso a bases de datos
 - Envío de correo electrónico
 - Generación dinámica de imágenes
 - ...
- Casi todas proporcionan un conjunto de objetos integrados accesibles desde los *scripts*

8

ASP, CFM, JSP y PHP (y III)

- Otras soluciones:
 - Componente ActiveX de Microsoft Access
 - CGI: C, Perl, *shell* del sistema operativo, etc.
 - Servlets (Java)
 - CSP (<http://www.intersystems.es/>): Caché ObjectScript y Caché Application Tags
 - Apache::ASP (<http://www.nodeworks.com/asp/>): Perl
 - Embperl (<http://perl.apache.org/embperl/index.html>): Perl
 - Mason (<http://www.masonhq.com/>): Perl
 - Lasso (<http://www.blueworld.com/>): Lasso Dynamic Markup Language
 - Zoppe (<http://www.zope.org/>): Python

9

ASP (I)

- Microsoft Active Server Pages
- `http://www.microsoft.com`
- Tecnología propietaria de Microsoft:
 - Sistemas operativos: Microsoft Windows 9x, Windows NT, Windows 2000 y Windows XP
 - Servidores web: Microsoft Personal Web Server e Internet Information Server

10

ASP (II)

- Hace uso de la tecnología de componentes de Microsoft: COM/DCOM
- Conectividad BD: OLE DB y ODBC
- Es un entorno (*framework*), no un lenguaje de programación → Se pueden emplear distintos lenguajes: VBScript y JavaScript (instalación estándar), PerlScript, etc.

11

ASP (III)

- Objetos integrados (intrínsecos):
 - Request: se emplea para recuperar la información enviada desde un formulario, a través de una URL o mediante cookies
 - Response: se emplea para enviar una respuesta al navegador (generar HTML o redirigir a otra dirección)
 - Session: se emplea para almacenar información persistente sobre un usuario (por ejemplo, variables locales)

12

ASP (IV)

- **Objetos integrados:**
 - `Application`: se emplea para almacenar información persistente sobre una aplicación (por ejemplo, variables globales)
 - `Server`: se emplea para crear instancias de componentes, como conexiones con bases de datos, envío de correo electrónico, etc.

13

ASP (V)

- Dispone del fichero `Global.asa`, con eventos que se ejecutan al iniciar (finalizar) una aplicación o una sesión
- Debido a su enorme popularidad y facilidad de uso, se ha “portado” a otros sistemas operativos como Linux (Chilisoft ASP)

14

ASP (VI)

- Ventajas:
 - El mejor sistema para programadores que conozcan Visual Basic
 - Fácil de aprender para los que no conozcan Visual Basic
 - Gratuito con el sistema operativo
 - Cientos de componentes (un gran mercado)
 - Muy extendido: miles de programadores, miles de sitios web → Una gran base de usuarios y de conocimiento

15

ASP (VII)

- Desventajas:
 - En la práctica, sólo para sistemas operativos Microsoft
 - La mayoría de los componentes no son gratuitos
 - Producto cerrado → Los fallos tardan en depurarse
 - Soporte múltiples lenguajes → Inherentemente más lento bajo las mismas condiciones

16

ASP (VIII)

- Ejemplo:

```
<html><body>
Fecha: <%= Date() %><br>
<%
  For i = 1 To 10
    Response.Write "Número: " & i & "<br>"
  Next
%>
</body></html>
```

17

ASP (y IX)

- Ejemplo:

```
<html><body>
<table>
<%
  Set con = Server.CreateObject("ADODB.Connection")
  con.Open "mibd"
  Set rst = con.Execute("SELECT * FROM mitabla")

  Do While Not rst.EOF
    Response.Write "<tr><td>" & rst("micampo1") &
"</td>"
    Response.Write "<td>" & rst("micampo2") &
"</td></tr>"
    rst.MoveNext
  Loop
%>
</table>
</body></html>
```

18

CFM (I)

- Macromedia (allaire) ColdFusion
- <http://www.macromedia.com>
- Disponible en:
 - Sistema operativo:
 - Microsoft Windows
 - Solaris
 - HP-UX
 - Linux
 - Servidor web:
 - Microsoft Internet Information Server y Personal Web Server
 - Apache
 - Netscape Enterprise Server
 - ...

19

CFM (II)

- Emplea etiquetas similares a HTML → ColdFusion Markup Language (CFML)
- Permite al programador definir etiquetas propias al estilo de HTML
- Puede emplear componentes en COM o CORBA
- Conectividad BD: ODBC y nativo (Sybase, Oracle, Informix, IBM DB2)
- Diversas versiones: Express (gratuita), Professional y Enterprise

20

CFM (III)

- Ventajas:
 - Muy sencillo de usar para la gente que sólo conozca HTML → Misma sintaxis
 - Existe etiquetas para la mayoría de las cosas que se necesitan hacer en una página web
 - Multiplataforma

21

CFM (IV)

- Desventajas:
 - Cómo se basa en etiquetas, es difícil crear sentencias de control complejas
 - “Extraño” para los programadores
 - Poco extendido

22

CFM (V)

- Ejemplo:

```
<html><body>
Fecha: <CFOUTPUT>#DateFormat(Now())#</CFOUTPUT><br>
<CFLOOP INDEX="i" FROM="1" TO="10">
Número: <CFOUTPUT>#i#</CFOUTPUT><br>
</CFLOOP>
</body></html>
```

23

CFM (y VI)

- Ejemplo:

```
<html><body>
<CFQUERY name="miconsulta" datasource="mibd">
SELECT * FROM mitabla
</CFQUERY>
<table>
<CFOUTPUT query="miconsulta">
<tr>
<td>#micampo1#</td><td>#micampo2#</td>
</tr>
</CFOUTPUT>
</table>
</body></html>
```

24

JSP (I)

- Sun JavaServer Pages
- <http://java.sun.com>
- Multiplataforma (basado en Java)
- JSP es la tecnología más moderna → Existen menos programadores, menos recursos
- Apareció para resolver los problemas que presentaban los *servlets* (pequeños programas en Java que se ejecutan en los servidores web → CGI's avanzados)

25

JSP (II)

- Una página JSP se traduce automáticamente a un servlet que posteriormente se compila y ejecuta
- Dispone de componentes: Java Beans y Enterprise Java Beans (EJB)
- Permite definir etiquetas propias al estilo de HTML
- Conectividad BD: JDBC
 - Puente JDBC-ODBC

26

JSP (III)

- **Objetos integrados (intrínsecos):**
 - `request`: se emplea para recuperar la información enviada desde un formulario, a través de una URL o mediante cookies
 - `response`: permite gestionar las cabeceras HTTP y las cookies
 - `out`: se emplea para enviar HTML al cliente

27

JSP (IV)

- **Objetos integrados:**
 - `session`: se emplea para almacenar información persistente sobre un usuario (por ejemplo, variables locales)
 - `application`: se emplea para almacenar información persistente sobre una aplicación (por ejemplo, variables globales)
 - **Otros:** `pageContext`, `config`, `page`, `exception`, etc.

28

JSP (V)

- Ventajas:
 - El mejor sistema para programadores que conozcan Java
 - Multiplataforma
 - Disponible toda la potencia de Java:
 - Orientado a objetos
 - Robusto
 - Excelente gestión de errores

29

JSP (VI)

- Desventajas:
 - Difícil para los que no conozcan Java (curva de aprendizaje grande)
 - Poco práctico para pequeños proyectos
 - Exige un diseño del sistema muy robusto y bien realizado

30

JSP (VII)

- Ejemplo:

```
<%@ page import = "java.util.Date" %>
<html><body>
<%
    Date hoy = new Date();
%>
Fecha: <%= hoy.toString() %><br>
<%
    for(int i = 1; i<= 10; i++)
        out.print("Número: " + i + "<br>");
%>
</body></html>
```

31

JSP (y VIII)

- Ejemplo:

```
<%@ page import = "todo lo necesario.*" %>
<html><body>
<table>
<%
    ResultSet rs = null;
    Statement stmt = null;
    String query = null;
    String sUrl = new String("jdbc:odbc:MiBD");

    Connection conexion = null;
    Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
    conexion = DriverManager.getConnection(sUrl,"usuario","contrassena");
    stmt = conexion.createStatement();
    query = "SELECT * FROM miTabla";
    rs = stmt.executeQuery(query);

    while(rs.next())
    {
        <tr><td><%=rs.getString("campoString")%></td>
        <td><%=rs.getInt("campoInteger")%></td></tr>
    }
%>
</table>
</body></html>
```

32

PHP (I)

- PHP
- `http://www.php.net`
- Iniciativa pública
- Código abierto
- Multiplataforma:
 - Diversos sistemas operativos
 - Diversos servidores web

33

PHP (II)

- Lenguaje de programación propio basado en Perl y C (C++ y Java)
- Amplia librería de funciones para hacer “casi todo”
- Conectividad BD: ODBC y nativa (Adabas, MySQL, Oracle, Informix, etc.)

34

PHP (III)

- Ventajas:
 - Gratuito
 - Multiplataforma
 - El más rápido
 - Mejor manejo de memoria
 - Fácil de aprender para los programadores que conozcan Perl y C (C++ y Java)
 - Se integra muy bien con Apache (servidor web) y MySQL (servidor de bases de datos) → Solución completa
 - Amplio soporte en Internet
 - Crecimiento muy rápido

35

PHP (IV)

- Desventajas:
 - Acceso a bases de datos → No está estandarizado (diferentes bases de datos tienen diferente sintaxis)
 - Lenguaje “confuso” en algunos aspectos (sensible mayúsculas/minúsculas: variables sí, funciones propias no)
 - No existen objetos intrínsecos (sesión, petición, etc.)
 - No existe una compañía detrás de él

36

PHP (V)

- Ejemplo:

```
<html><body>
Fecha: <?= date("d/m/Y") ?><br>
<?
    for($i = 1; $i <= 10; $i++)
        echo "Número: " . $i . "<br>";
?>
</body></html>
```

37

PHP (y VI)

- Ejemplo:

```
<html><body>
<table>
<?
    $conexion = odbc_connect("miBD", "usuario", "contrassena");
    $query = "SELECT * FROM miTabla";
    $resultados = odbc_exec($conexion, $query);

    while (odbc_fetch_row($resultados))
    {?>
        <tr><td><?=odbc_result($resultados, "miCampo1")?></td>
        <td><?=odbc_result($resultados, "miCampo2")?></td></tr>
    <?
    }
?>
</table>
</body></html>
```

38

Programación en Internet – Curso 2007-2008

Comparativa (I)

	ASP	CFM	JSP	PHP
Facilidad de uso				
Dificultad aprendizaje				
Soporte de BD				
Multiplataforma				

39

Programación en Internet – Curso 2007-2008

Comparativa (y II)

	ASP	CFM	JSP	PHP
Coste inicial				
Coste medio				
Rendimiento				
Escalabilidad				

40

Programación en Internet – Curso 2007-2008

Estadísticas de uso

	Octubre 2000	Marzo 2002	Crecimiento	Share Marzo 2002
ASP	3.166.710	11.958.185	278%	48%
CFM	936.223	4.950.133	429%	20%
JSP	24.435	413.827	1594%	2%
PHP	157.470	7.549.230	4694%	30%

41