

Tema 5

Tecnologías web

Antonio Sanz – ansanz@unizar.es

Comercio Electrónico



Tecnologías web

Índice

- Programación dinámica
- PHP
- Javascript
- XML





Tecnologías web

Programación web dinámica





Tecnologías web

Programación dinámica

- Hasta ahora → Desarrollo de páginas web estáticas + formularios web (capacidad de recoger información)
- Objetivo → Ser capaces de procesar esa información y responder al usuario
- Solución → Programación dinámica



Tecnologías web

Programación dinámica

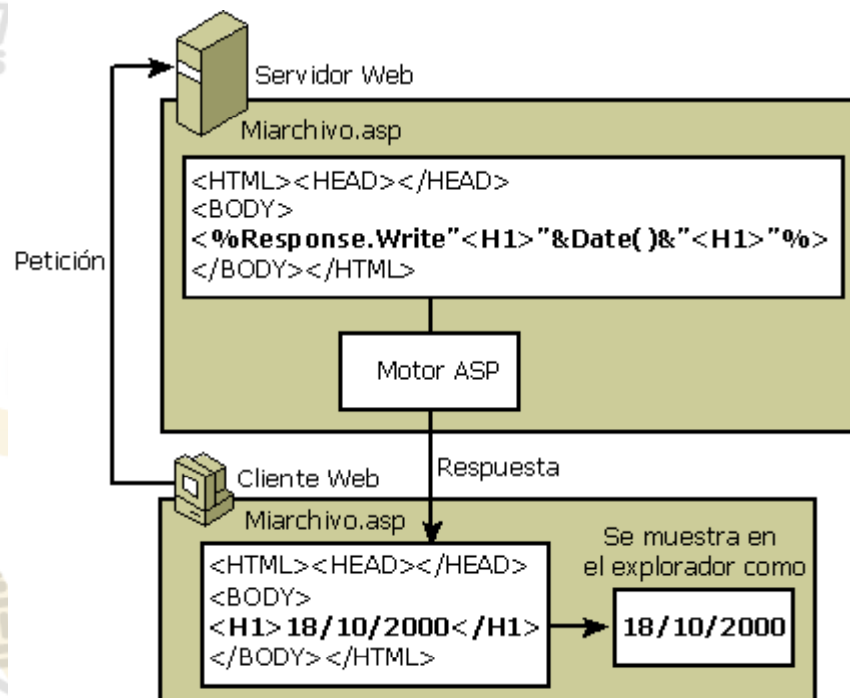
■ Principales lenguajes empleados a día de hoy:

- PHP
- ASP & ASP.NET
- Java & JSP



Tecnologías web

Programación dinámica



Funcionamiento :

1. Conexión
2. El servidor le entrega el fichero al motor
3. El motor ejecuta el código y entrega un resultado al servidor
4. El servidor devuelve el resultado al cliente



Tecnologías web

Programación dinámica

■ PHP (PHP Hypertext Processor)

- Lenguaje de propósito general especialmente orientado a web
- Lenguaje interpretado, no asociado a ninguna plataforma o sistema operativo
- Excelentes capacidades de generación de código HTML y conexión a bases de datos (excelentes resultados con MySQL)
- Gran cantidad de aplicaciones open source desarrolladas con este lenguaje



Tecnologías web

Programación dinámica

■ ASP (Active Server Pages)

- Lenguaje creado por Microsoft para el desarrollo de aplicaciones web
- Lenguaje interpretado, con clases para generación de código HTML y conexiones a bases de datos
- En la actualidad, en desuso → Actualización a ASP.NET



Tecnologías web

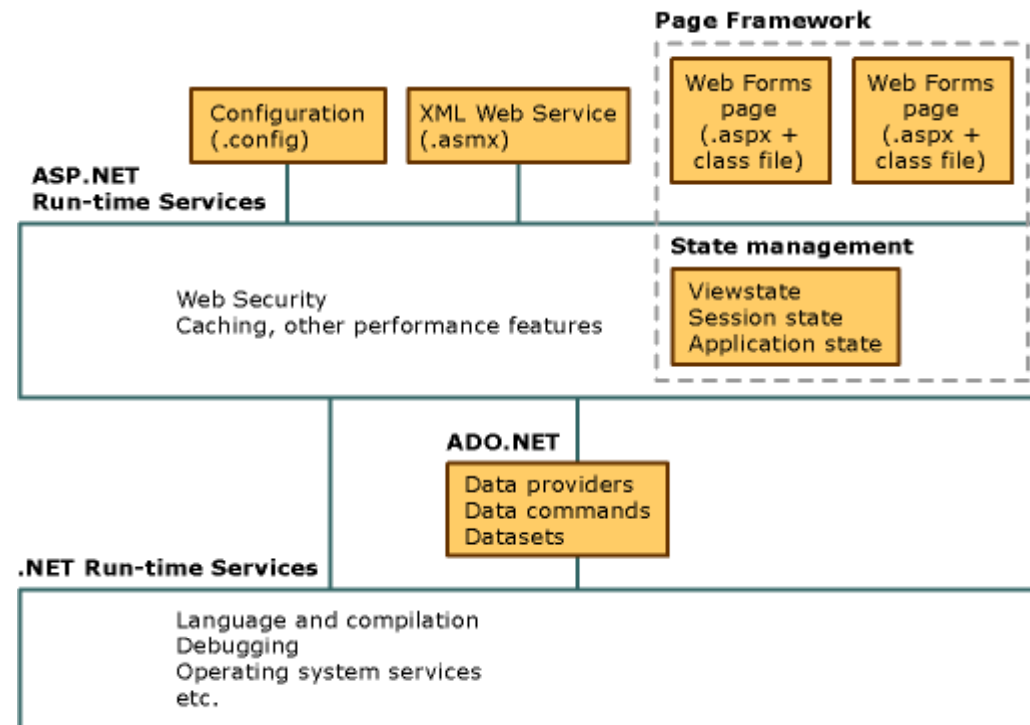
Programación dinámica

- **.NET Framework** → Plataforma de desarrollo multipropósito de Microsoft
- ASP.NET → Componente de la plataforma orientado a aplicaciones web
- Utiliza ADO.NET para la conexión a bases de datos
- Permite generar código en cualquier lenguaje .NET (C#, Visual Basic, etc.)
- Permite emplear las clases .NET del sistema operativo



Tecnologías web

Programación dinámica



Tecnologías web

Programación dinámica

■ ¿Cómo funciona el .NET?

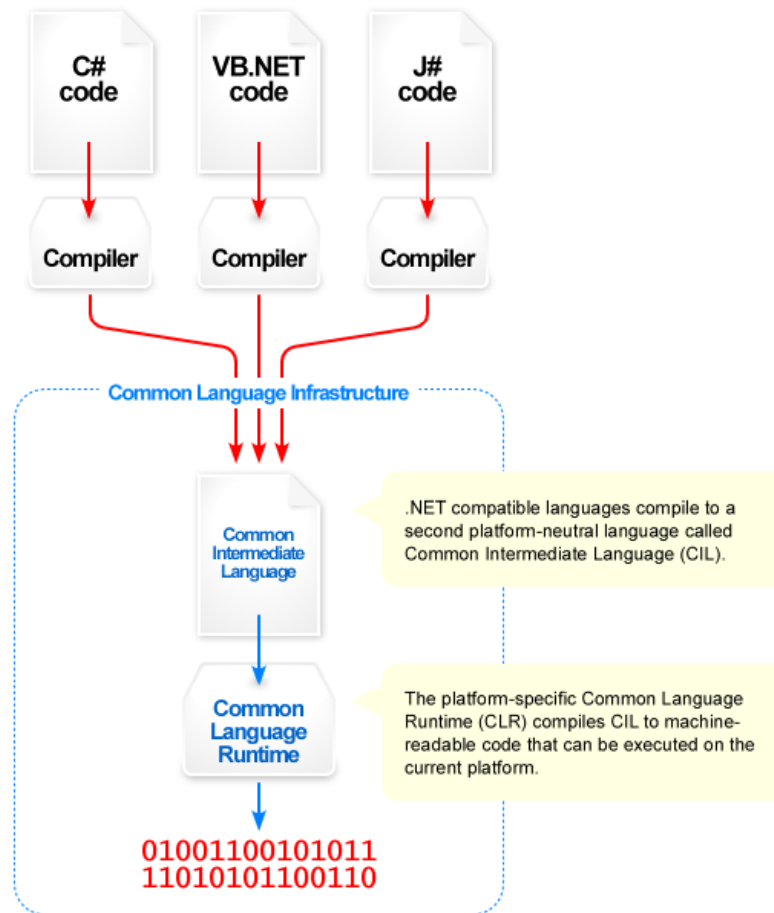
- Creamos un componente en el lenguaje que deseemos (C#, Visual Basic, J#)
- Lo compilamos a una base común → CIL (Common Intermediate Language)
- Luego se “recompila” este CIL con un CLR (Common Language Runtime) para adaptarlo a cada plataforma



Tecnologías web

Programación dinámica

Programando en .NET



Tecnologías web

Programación dinámica

■ Ejemplo:

```
<html>  
<body>  
<center> <h2>Hello W3Schools! </h2>  
<p> <%Response.Write(now())%> </p>  
</center>  
</body>  
</html>
```



Tecnologías web

Programación dinámica

- Java & páginas web: Múltiples posibilidades:
 - JSP (Java Server Pages)
 - Servlets
 - EJB (Enterprise Java Beans)
- Lenguaje compilado, necesita de un runtime (en Apache → Tomcat)



Tecnologías web

Programación dinámica

■ JSP (Java Server Pages):

- Permite introducir trozos de código Java en páginas web
- Estas páginas son luego compiladas y se genera un código Java → Servlet
- Ventajas: Fácil de programar (se usa en cosas sencillas)



Tecnologías web

Programación dinámica

Ejemplo: Página JSP

```
<%@ page errorPage="myerror.jsp" %>
<%@ page import="com.foo.bar" %>
<html> <head>
<%! int serverInstanceVariable = 1;%> ...
<% int localStackBasedVariable = 1; %>
<table>
<tr><td><%= toStringOrBlank( "expanded inline data " + 1 )
    %></td></tr>
...

```





Tecnologías web

Programación dinámica

- Servlets: Código Java puro que se ejecuta dentro de un “contenedor”
- El HTML es generado de forma dinámica por el servlet
- Ventajas: Mucho más potentes que los JSP (que son una subdivisión de los servlets)
- Pueden ser empleados para crear servicios web



Tecnologías web

Programación dinámica

- Enterprise Java Beans: Java con esteroides
- Ofrece persistencia, concurrencia, seguridad → Toda la lógica de negocio
- Mucho más complejos de programar, pero nos dan muchísima potencia



Tecnologías web

Programación dinámica

- Modelo de programación importante en web → MVC (Model View Controller)
 - View → Visualiza la información → HTML
 - Controller → Controla los procesos de negocio → Lenguaje de programación
 - Model → Tiene el modelo del negocio, la información → Base de datos



Tecnologías web

Programación dinámica

- En arquitecturas pequeñas, todos los servicios se ejecutan en un mismo servidor
- En arquitecturas grandes, se separan los servicios (sólo si se sigue bien el modelo)
- Ejemplo: capa de presentación + capa de negocio + capa de datos → Arquitectura “multitier”



Tecnologías web

Programación dinámica

- Desarrollar aplicaciones con PHP:
 - IDE: Eclipse + PDT (Perl Development Tools)
 - Framework: CakePHP/Zend PHP Framework
 - Para cosas pequeñas: PHPDesigner o PHP IDE





Tecnologías web

Programación dinámica

■ Desarrollar aplicaciones en .NET:

- IDE → Visual Studio .NET (ya viene con el framework puesto)
- Alternativa open source a .NET → MONO (MonoDevelop)



Tecnologías web

Programación dinámica

- Desarrollar aplicaciones en Java:
 - Sun Java Studio
 - Eclipse JDT (Java Development Toolkit)
 - NetBeans



Tecnologías web

Conclusiones

- Existen multitud de tecnologías para el desarrollo de aplicaciones web
- Se debe elegir la más adecuada a cada tipo de proyecto (coste, necesidades, experiencia, recursos disponibles y tecnologías ya existentes)
- Seguir una buena metodología (MVC) y elegir un buen IDE + Framework



Tecnologías web

PHP





Tecnologías web

PHP

- Lenguaje interpretado orientado a la creación de aplicaciones web
- Muy sencillo en su base, pero potente (soporta programación orientada a objetos)
- ¿Qué sabe hacer bien? → Generar código HTML y conectarse a bases de datos (a Mysql, instantáneo)



Tecnologías web

PHP

■ Creación de páginas PHP:

- Inserción de “trozos de código PHP” dentro del HTML
- Inicio de PHP → `<?php`
- Fin de PHP → `?>`
- Ejemplo: `<?php echo “Hola Mundo”; ?>`



Tecnologías web

PHP

■ Ejemplo de página en PHP:

```
<html>
<head>
<title>Ejemplo de PHP</title>
</head>
<body>
<?php echo "Hola Mundo" ; ?> <br>
</body>
</html>
```



Tecnologías web

PHP

■ Sintaxis básica :

- Todas las líneas terminan con “;”
- Las variables se definen con: \$variable
- No existen tipos de variables (enteros, strings son iguales)
- Comentarios: // Esto es un comentario, /* Esto vuelve a ser un comentario */



Tecnologías web

PHP

■ Operaciones básicas:

■ Sacar salida por pantalla:

```
echo "Esta es mi $salida <br>";  
echo "<a href='link.htm'>$mi_enlace</a>";
```

■ Comparaciones:

```
$c == "5" ; $c!= $b ; $a && $b; $a || $b;
```

■ Aritméticas:

```
$c = $a + $b + 5 ; $c++ ; $c = $c * 2 ;
```

■ Otras operaciones interesantes:

```
$nombre_completo = $nombre . $apellido1 . $apellido2  
$cadena_limpia = trim($cadena_con_saltos_de_linea_y_espacios_al_final)
```



Tecnologías web

PHP

■ Control de flujo:

Estructura de control “IF” :

```
if ( $stock != 0) {  
    echo “Tenemos productos”;  
}  
else {  
    echo “Lo sentimos, no tenemos más productos”;  
}
```

Estructura de control “FOR”:

```
for ( $i = 1: $i <= $numero_productos ; $i++ ) {  
    echo “Producto numero $i ”;  
}
```

Estructura de control “WHILE”:

```
while ( $i < $numero_productos ) {  
    echo “Producto número: $i ” ;  
    $i++ ;  
}
```



Tecnologías web

PHP

■ Generación de código HTML:

```
echo "<html><body>Hola  
Mundo</body></html>";
```

- Nota: PHP tiene las comillas dobles para su lenguaje
→ dentro hay que usar comillas simples:

```
echo "<a href=\"link.htm\">$mi_enlace</a>"; ← Mal
```

```
echo "<a href='link.htm'>$mi_enlace</a>"; ← Bien
```



Tecnologías web

PHP

- Recoger variables de formularios web:

```
$email = $_HTTP_POST_VARS ['nombre'] ;
```

\$email = Variable del PHP

'nombre' = Etiqueta "name" del HTML

- Si el método es GET → HTTP_GET_VARS

- Consejo: Misma nomenclatura en HTML que en PHP



Tecnologías web

PHP

■ Conexiones a bases de datos:

// Definimos los parámetros

\$host = "localhost";

\$user = "miusuario";

\$password = "micontraseña";

\$database = "minegocio";

// Nos conectamos a la BD

\$conexion = mysql_connect(\$host, \$user, \$password, \$database);

// Preparamos la consulta (ojo a la concatenación con el . para dar claridad)

\$consultaSQL = "select ";

\$consultaSQL . = " id_producto";

\$consultaSQL . = ", nombre";

\$consultaSQL . = ", precio";

\$consultaSQL . = ", stock";

\$consultaSQL . = "FROM \$TABLA_DATOS";

\$consultaSQL . = " ORDER BY precio" ;



Tecnologías web

PHP

// Realizamos la consulta, y los datos se almacenan en una estructura que toma // la forma de la variable \$resultado

`$resultado = mysql_query($consultaSQL);`

// Obtenemos el número de resultados

`$numFilas = mysql_num_rows($resultado);`

// Recogemos los resultados. Estos se almacenan en una matriz de forma que // cada fila tiene los resultados de un producto, y cada columna los resultados // de una categoría pedida.

// Con esto nos hacemos con los resultados del producto más caro

// La idea es meter este código en un bucle que vaya procesando todos los // productos

`$id = mysql_result($resultado, 0, 0);`

`$nombre = mysql_result($resultado, 0, 1);`

`$precio = mysql_result($resultado, 0, 2);`

`$stock = mysql_result($resultado, 0, 3);`

// Cerramos la conexión y liberamos recursos

`mysql_free_result($resultado);`

`mysql_close();`



Tecnologías web

PHP

■ Recursos de PHP:

- Manual de PHP

<http://www.php.net/docs.php>

- PEAR (PHP Extension and Application Repository)

<http://pear.php.net/>

- PHP Classes Repository

<http://www.phpclasses.org/>

- O'Really PHP Dev Centre

<http://www.onlamp.com/php/>



Tecnologías web

Conclusiones

- PHP: Lenguaje sencillo y potente, indicado para aplicaciones pequeñas y medianas
- PHP básico + PHP avanzado (oop)
- Gran cantidad de código ya generado
- Una buena opción para el desarrollo de aplicaciones web





Tecnologías web

Javascript

Javascript



Tecnologías web

Javascript

- **Javascript** → Lenguaje de scripting diseñado para ofrecer una cierta interactividad en la web
- Lenguaje interpretado
- Se inserta dentro del código HTML
- ¡¡Java NO es Javascript!!
- Uso más directo → Mejorar la usabilidad de una aplicación web



Tecnologías web

Javascript

- ¿Qué podemos hacer con Javascript?
 - Poner código en páginas HTML
 - Reaccionar a eventos del cliente (imágenes que cambian cuando pasamos el ratón por encima)
 - Crear cookies
 - Validar formularios
 - Detectar el navegador y la resolución de la pantalla de un cliente (y cargarle una página)



Tecnologías web

Javascript

- Lenguaje de programación DESDE EL CLIENTE (no podemos interaccionar con el servidor)
- Se ejecuta en el cliente → Muy limitado por temas de seguridad
- En un código HTML, determinado por las etiquetas `<script>` y `</script>`



Tecnologías web

Javascript

- Ejemplo de una página web con Javascript:

```
<html>  
<body>  
<script type="text/javascript">  
    document.write("Hello World!")  
</script>  
</body>  
</html>
```



Tecnologías web

Javascript

- Javascript puede manipular los objetos creados en HTML (y los suyos):

Table 1-1 Creating and Working with Objects		
<i>Object</i>	<i>HTML Tag</i>	<i>JavaScript</i>
Web page	<BODY> . . . </BODY>	document
Image		document.my Image
HTML form	<FORM name="my Form"> . . . </FORM>	document.my Form
Button	<INPUT TYPE="button" NAME="my Button">	document.my Form. my Button



Tecnologías web

Javascript

■ Dónde colocar los scripts:

- En el <head> si queremos que estén disponibles en toda la página
- En el <body> si queremos que se ejecuten cuando se carga la página
- En un fichero externo para la reutilización de código:

```
<script src="xxx.js"> </script>
```



Tecnologías web

Javascript

■ Sintáxis básica de JavaScript:

- No necesitamos terminadores de línea (bye ;)
- Comentarios: // Esto es un comentario
- Variables: var nombre; var nombre = "Pepe"
- Javascript es "case sensitive" → var pepe/var Pepe ¡son diferentes!



Tecnologías web

Javascript

- Escribir texto en pantalla:

`Document.write("Esto sale en la propia página web")`

- Crear ventanas de confirmación: `confirm()`

`Confirm("¿Está seguro de que quiere hacer esto?")`

- Crear popups: `alert()`

`alert("Esta es una ventana de aviso")`



Tecnologías web

Javascript

- Condicionales:

if (*condition*)

{ *code to be executed if condition is true* }

else

{ *code to be executed if condition is not true* }

- Bucles for:

for (var=startvalue;var<=endvalue;var=var+increment)

{ *code to be executed* }

- Bucles while:

while (var<=endvalue) { *code to be executed* }



Tecnologías web

Javascript

- **Eventos** → Interacción del usuario con la página web (pasar el ratón, aceptar un formulario, etc ...)
- Javascript permite ejecutar código cuando se cumpla un evento:

```
<form method="post" action="xxx.htm" onsubmit="return  
checkForm()">
```

```
<input type="text" size="30" id="email"  
onchange="checkEmail()">
```

```
<a href="http://www.w3schools.com"  
onmouseover="alert('An onMouseOver event');return  
false">  </a>
```



Tecnologías web

Javascript

- **Objetos:** Javascript es un lenguaje orientado a objetos
- Puede emplear tanto los objetos HTML como sus propios objetos (y algunos más del navegador como window, screen, location)

```
var txt="Hello World!"  
document.write(txt.length)
```

- Objetos HTML → DOM (Document Object Model)

```
document.body.style.background="#FFCC80"
```



Tecnologías web

Javascript

```
<script type="text/javascript">
// Define a variable called cssName and a message
// called resolutionInfo
var cssName;
var resolutionInfo;
// If the width of the screen is less than 650 pixels
if( screen.availWidth < 650 ) {
// define the style Variable as the low-resolution style
cssName = 'lowres.css';
// Or if the width of the screen is less than 1000 pixels
} else
{
    if( screen.availWidth > 1000 ) {
// define the style Variable as the high-resolution style
cssName = 'highres.css';
// Otherwise
    } else {
// define the style Variable as the mid-resolution style
cssName = 'lowres.css';
    }
}
document.write( '<link rel="StyleSheet" href="" +
cssName + "" type="text/css" />' );
</script>
```



Tecnologías web

Javascript

■ Desarrollo con JavaScript:

- IDE → Aptana, Jside
- Frameworks → Prototype , Scriptaculous
- Debugging → FireBug (extensión de Firefox)



Tecnologías web

Javascript

■ Recursos adicionales:

- Tutorial de Javascript

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

- Referencia de Javascript

<http://docs.sun.com/source/816-6408-10/>

- Javascript.com – Recursos extra

<http://www.javascript.com/>



Tecnologías web

Conclusiones

- Javascript → Lenguaje que se ejecuta en el cliente
- Permite modificar páginas web sin tener que usar programación de servidor
- Interesantes posibilidades para “facilitar” la vida al usuario → Usabilidad



Tecnologías web

XML



Tecnologías web

XML

- **XML → Extensible Markup Language**
- Objetivo → Describir datos
- Las etiquetas son libres → “Háztelo tú mismo”
- Requiere de un DTD o un XML Schema → Forma de representar los datos



Tecnologías web

XML

- XML & HTML → No son exclusivos
 - HTML → Mostrar datos (formato)
 - XML → Describir datos (dar significado)
- Son perfectamente compatibles (de hecho, XML es EL standard para transmitir información)
- AJAX, SOAP, Web Services = HTML + XML



Tecnologías web

XML

- Uso principal del XML → Intercambio de información entre **diferentes** sistemas de información
- Ejemplo: Aplicación Java en un servidor Windows “habla” con un PHP en un servidor Linux
- Es el EDI (Electronic Data Interexchange), pero funcional



Tecnologías web

XML

■ Ejemplo de documento XML:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persona>
  <nombre>Jose</nombre>
  <apellido1>Lopez</apellido1>
  <apellido2>Martinez</apellido2>
  <email>jlopez@unizar.es</email>
  <organizacion>Unizar</organización>
</persona>
```



Tecnologías web

XML

■ Reglas del XML:

- Todos los elementos tienen que tener etiquetas de cierre
- Todos los documentos tienen que tener un elemento raíz
- Todos los elementos tienen que ir en minúsculas
- Todos los elementos tienen que estar correctamente anidados
- Todos los atributos tienen que estar señalados con comillas



Tecnologías web

XML

- Característica del XML → Extensible

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<persona>
  <nombre>Jose</nombre>
  <apellido1>Lopez</apellido1>
  <apellido2>Martinez</apellido2>
  <email>jlopez@unizar.es</email>
  <organizacion>Unizar</organización>
  <telefono>976765845</telefono>
  <facultad>CPS</facultad>
</persona>
```

- Esta definición EXTIENDE la anterior → ¡OK!



Tecnologías web

XML

- Los elementos XML pueden contener atributos:

```
<note date="12/11/2002"> ....
```

- ... pero es mejor definirlos como datos:

```
<note>  
<date>12/11/2002</date> ....
```

- ... y mejor todavía ser detallistas:

```
<note>  
<date>  
<day>12</day>  
<month>11</month>  
<year>2002</year> ....
```



Tecnologías web

XML

- Validación de los XML → Fundamental (recordad que el intercambio es entre sistemas, no entre personas) → ¡Hay que ser exactos!
- Un XML válido cumple un DTD (Document Type Definition) o un XML Schema → Formas de definir la información válida
- Ejemplo: `<?xml version="1.0" encoding="ISO-8859-1"?> <!DOCTYPE note SYSTEM "personas.dtd">`



Tecnologías web

XML

- Ejemplo (DTD):

```
<?xml version="1.0"?>
<!DOCTYPE persona [
  <!ELEMENT persona (nombre, apellido1, apellido2,email,
    organizacion)>
  <!ELEMENT nombre (#PCDATA)>
  <!ELEMENT apellido1 (#PCDATA)>
  <!ELEMENT apellido2 (#PCDATA)>
  <!ELEMENT email (#PCDATA)>
  <!ELEMENT organizacion (#PCDATA)>
]>
```

- ¿Para qué sirven? → Para que dos entidades se **pongan de acuerdo** en usar un formato común



Tecnologías web

XML

- ¿Cómo presentar datos con XML? → XSL (Extensible StyleSheet Language) → las CSS del XML
- ¿Cómo extraer datos de un XML? → XML Parser
- Ejemplo de XML → Feed RSS



Tecnologías web

XML

- **RSS: Really Simply Syndication**
- Es una forma sencilla de compartir contenido y de automatizar su distribución
- Emplea XML para su definición
- Ejemplo de uso de RSS → Sitios de noticias
- Lectores de FeedRss → FeedReader, Owl



Tecnologías web

XML

- Desarrollar XML:
 - Oxygen → Plugin para Eclipse
 - Propio WST (Web Standard Tools) de Eclipse
 - XMLBuddy



Tecnologías web

Conclusiones

- XML → Define la información, no cómo visualizarla
- Permite intercambiar información entre sistemas heterogéneos
- Es la piedra base de servicios de más alto nivel → AJAX & Web Services
- Tecnología con MUCHO futuro



Tecnologías web

¿Programamos alguna duda?

¿?

