

Lenguajes para la Web

(marcado y presentación)

Francisco José Suárez Alonso

UNIVERSIDAD DE OVIEDO

Área de Arquitectura y Tecnología de Computadores

Curso 2010/2011

Introducción



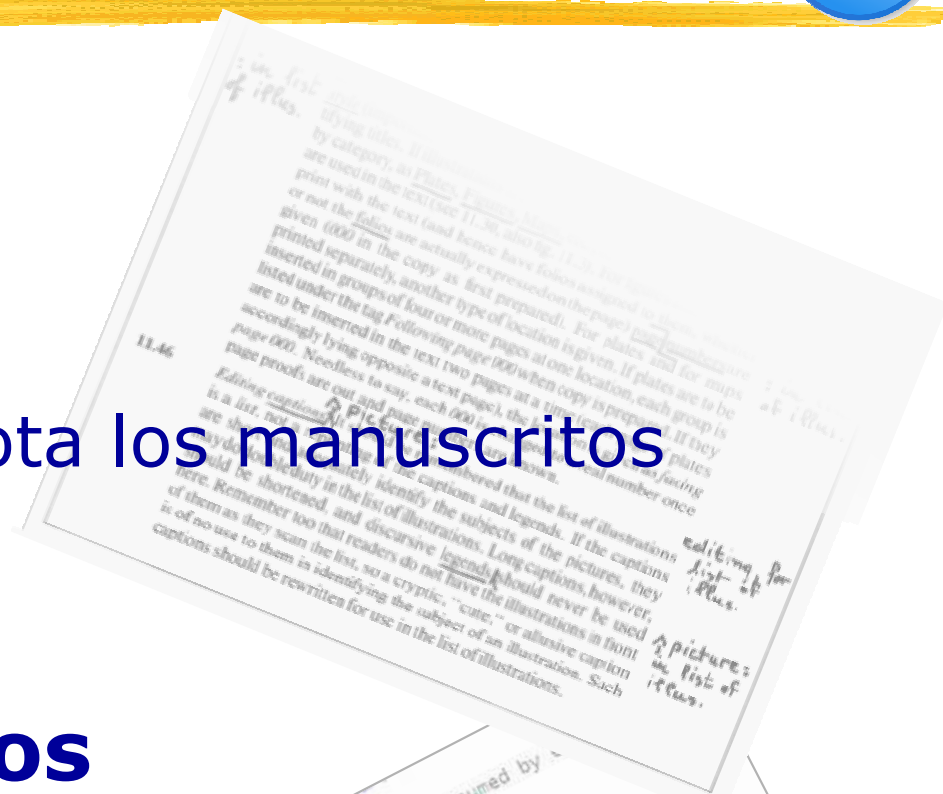


¿Qué es el marcado?

Documentos en papel

Marcado:

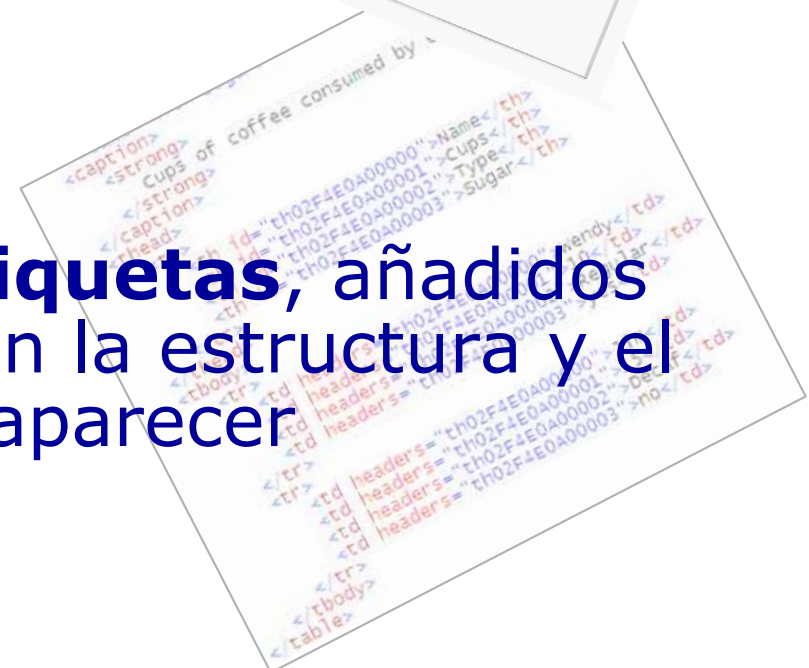
Forma en que el editor anota los manuscritos



Documentos electrónicos

Marcado:

Códigos, también llamados **etiquetas**, añadidos al texto electrónico que definen la estructura y el formato en el que tienen que aparecer





Define dos estructuras

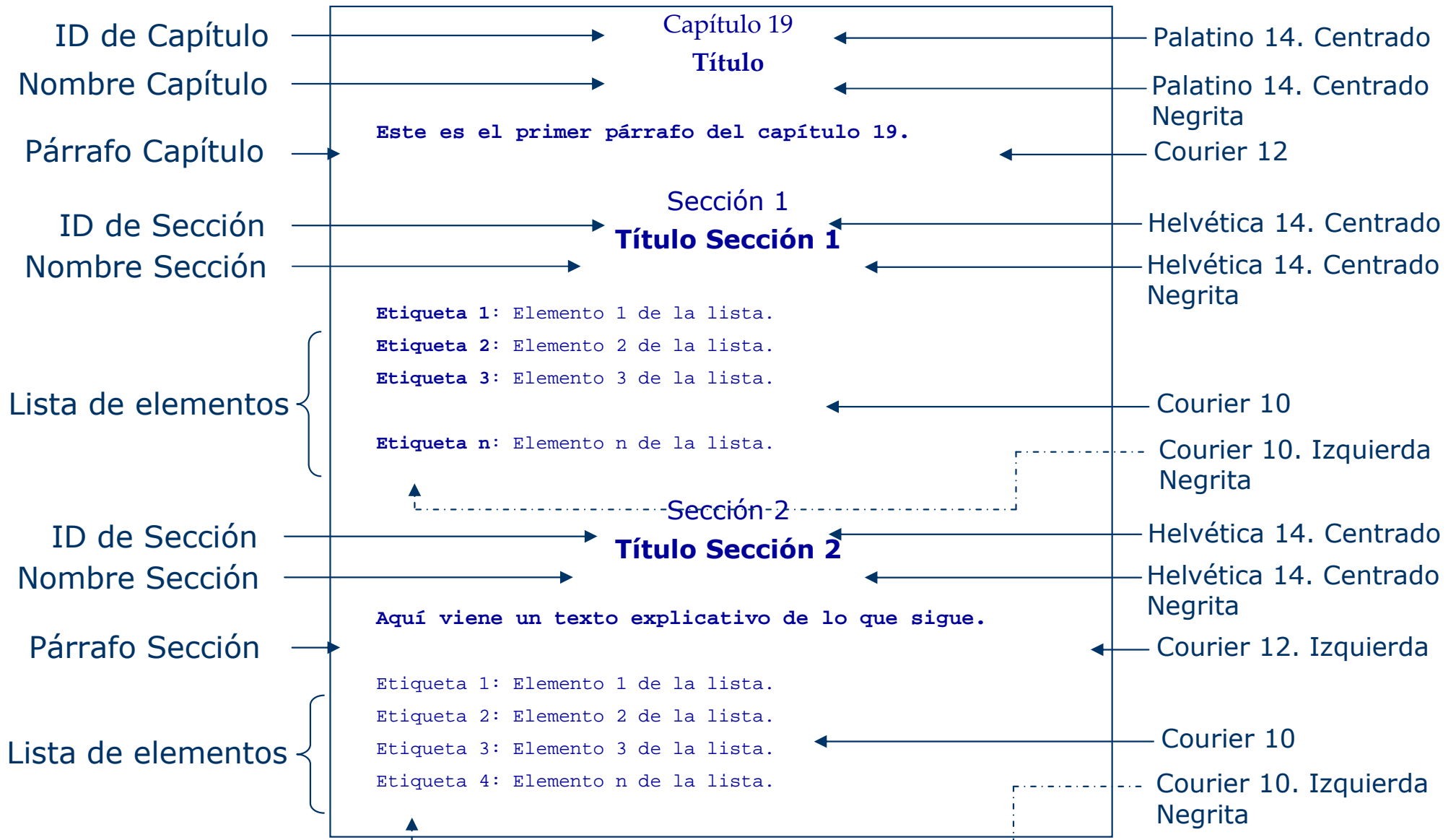
Estructura lógica: formada por las partes que lo componen y sus relaciones

Estructura física: indica la apariencia del documento, incluyendo sus componentes físicos, posicionamiento de elementos y tipografía empleada





Ejemplo

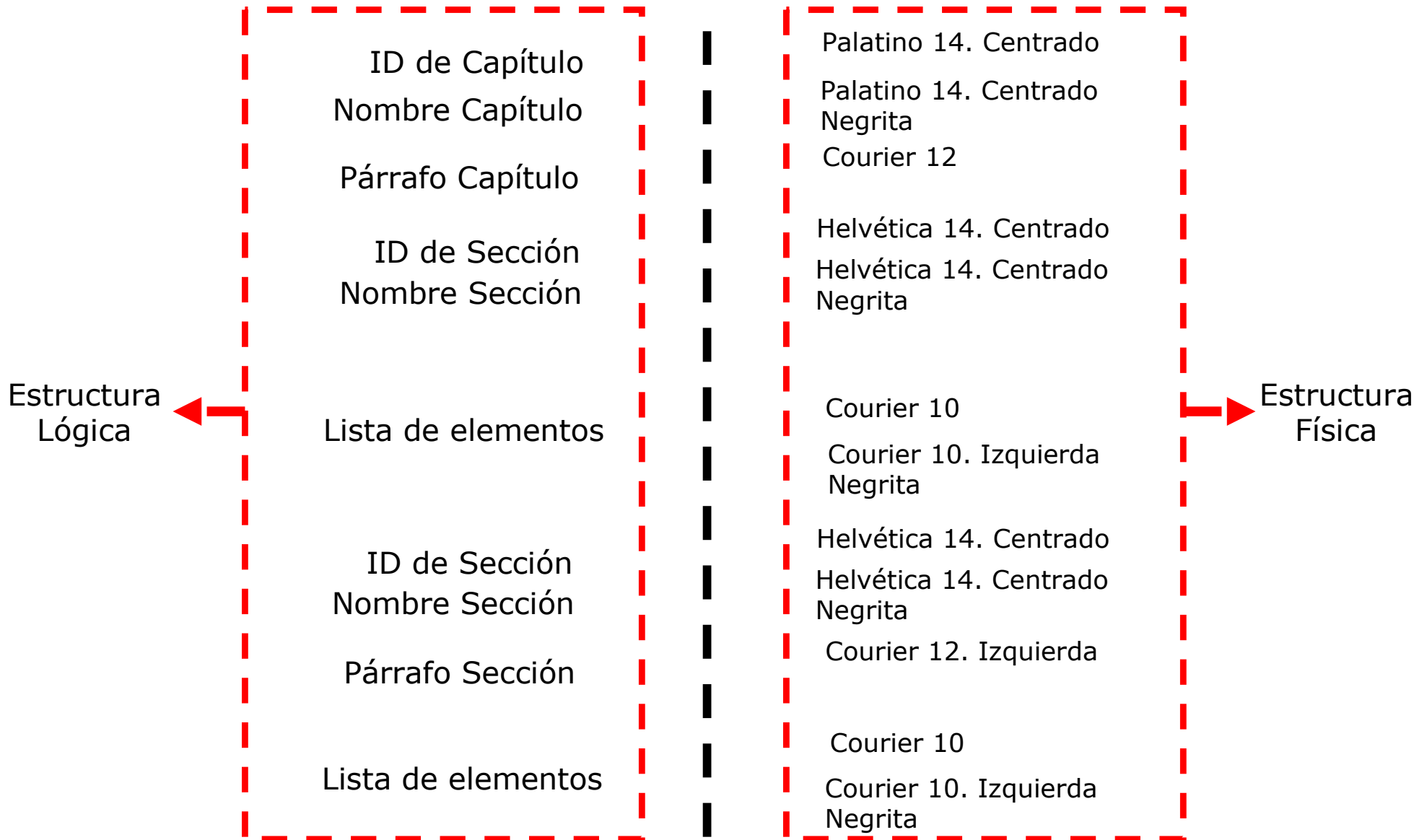


Estructura Lógica

Estructura Física



Ejemplo (continuación)



Lenguajes de marcado

Específico:



Genérico:



World Wide Web

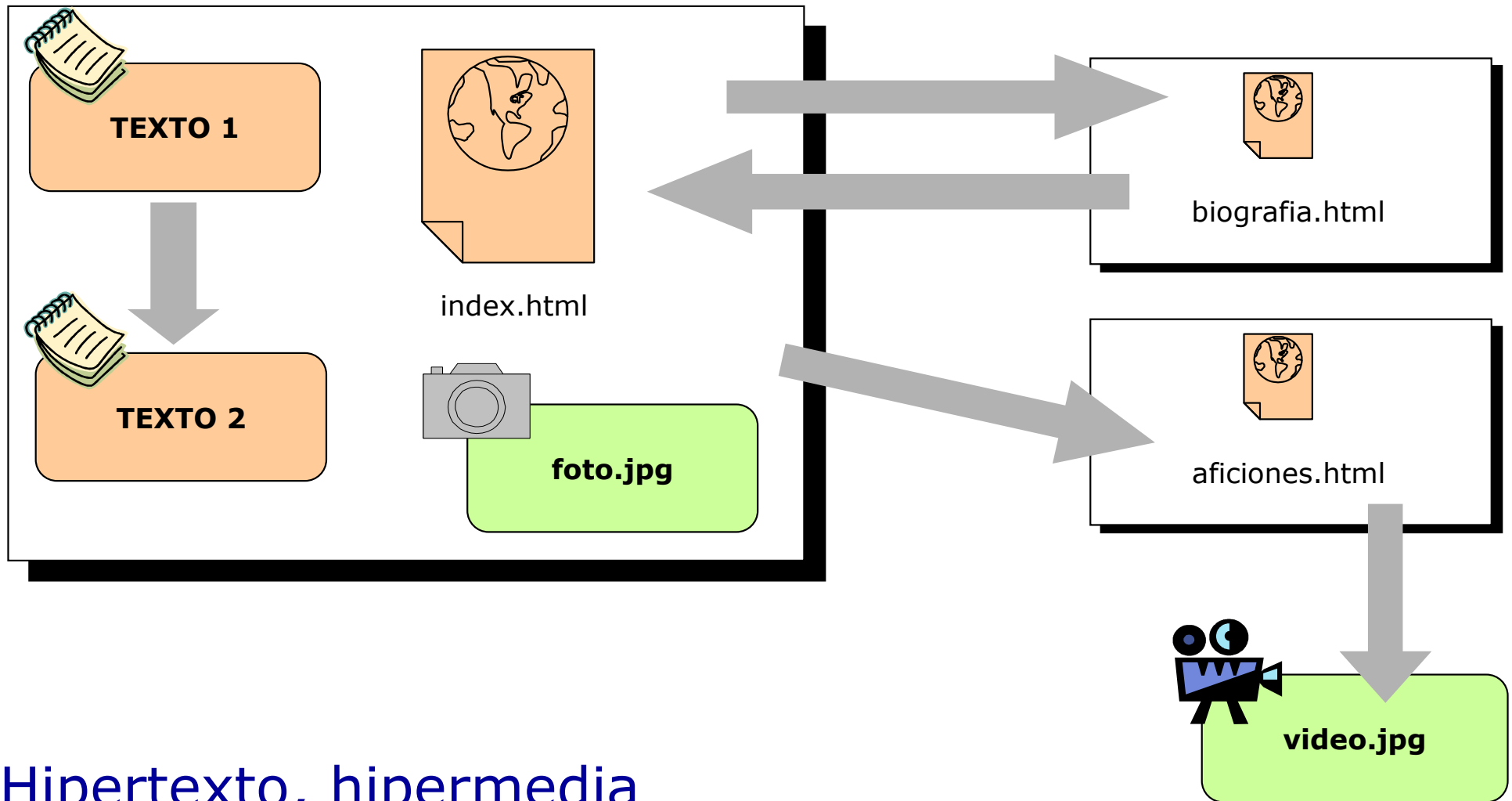




- WWW: ***World Wide Web***
- Sistema de información universal que utiliza:
 - ✓ Lenguaje de marcas HTML
(HyperText Markup Language)
 - ✓ Protocolo HTTP (HyperText Transfer Protocol)
- Hipertexto: Estructura de aplicaciones en nodos
 - ✓ Nodo: incorpora una serie de contenidos
 - ✓ Recorrido no lineal a través de hiperenlaces
- Sistema hipermedia: hipertexto + multimedia



Arquitectura del hipertexto



Hipertexto, hipermedia



Servidor

Cliente



Estructura del HTML



```
<html>
  <head>
    <title>Mi página personal</title>
  </head>
<body>
  <p>
    <h1>Página personal de Nacho Marín.</h1><br>
    <a href="biografia.html">Mi biografía</a>
    <a href="aficiones.html">Mis aficiones</a>
  </p>
  <p id="texto1">
    La multimedia tiene un montón de <a href="#aplicaciones">aplicaciones</a>.
    <br> . . .
  </p>
  
  <p id="texto2">
    Entre las aplicaciones de la multimedia nos encontramos con:
    <a name="aplicaciones">
      <ul>
        <li>Sistemas educativos.</li>
        <li>Presentaciones interactivas.</li>
        <li>etc.</li>
      </ul>
    </a>
  </p>
</body>
</html>
```

index.html

Hipervínculo a biografía y aficiones

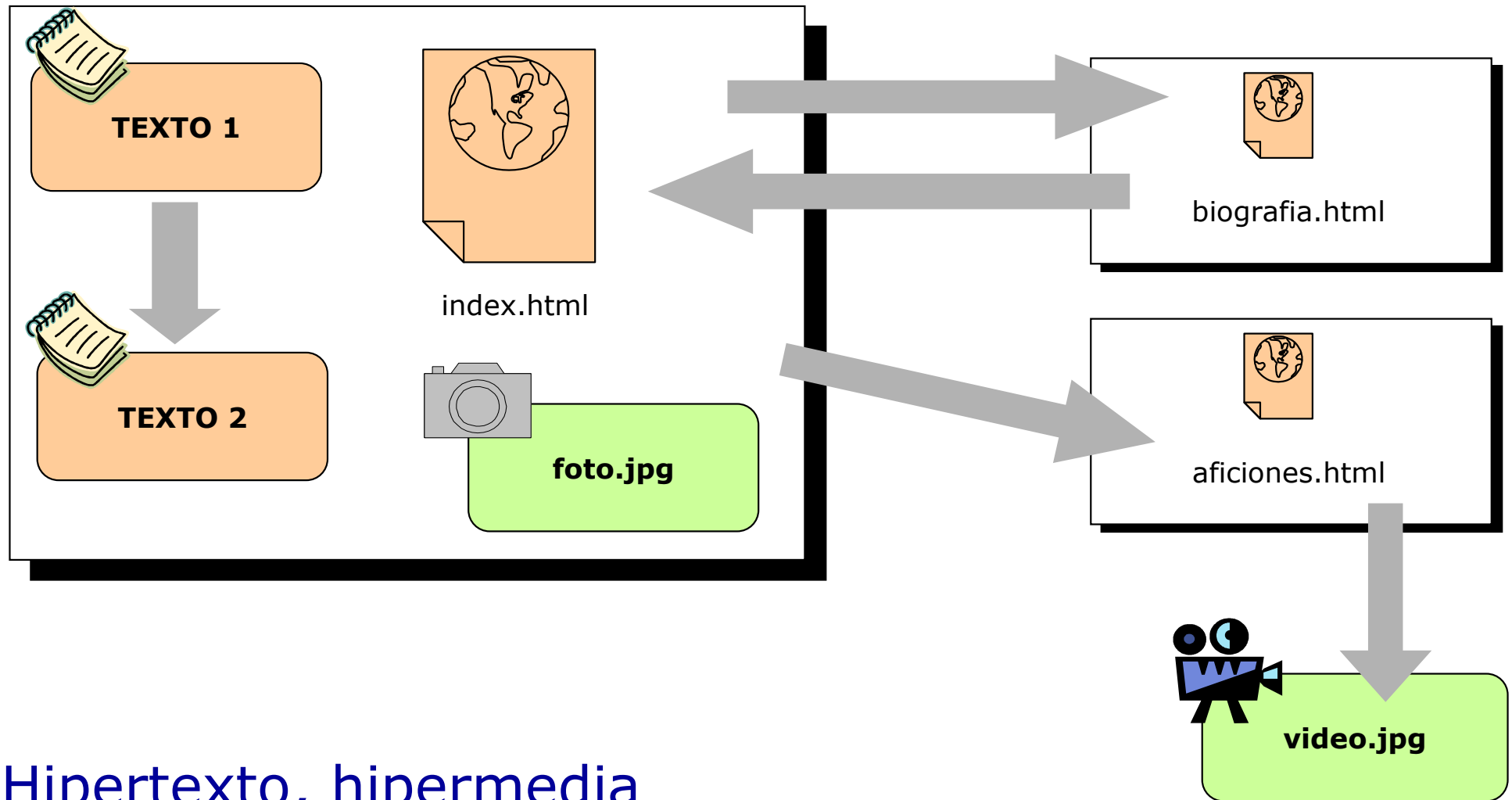
Párrafo texto 1

Foto

Párrafo texto 2



Arquitectura del hipertexto



Hipertexto, hipermedia

Estructura del HTML



```
<html>
  <head>
    <title>Biografía de Nacho Marín</title>
  </head>
  <body>
    <h1>Datos biográficos</h1>
    <h2>Nombre: Ignacio Marín</h2>
    <p>
      Nací el ....
    </p>
    <a href="index.html">Volver a la página principal</a>
  </body>
```

biografia.html

Enlace de retorno a página principal

```
<html>
  <head>
    <title>Aficiones</title>
  </head>
  <body>
    <h1>Deportes</h1>
    <p>
      Me encantan todos los deportes pero se me dan todos fatal.
      <a href="video1.mpg">Aquí</a> se me puede ver echando un partidín de basket.
    </p>
  </body>
</html>
```

aficiones.html

Enlace a un vídeo externo

Concepto: ancla (anchor)



Es un fragmento de texto "clicable" que actúa como origen o destino de un enlace:

✓ Páginas HTML, imágenes, vídeos o puntos concretos dentro de una página

- Elemento `<a>` en HTML
- Si actúa como origen, especifica su destino con el atributo **href**
- Si actúa como destino, especifica su origen con el atributo **name**



Estructura del HTML



```
<html>
  <head>
    <title>Mi página personal</title>
  </head>
<body>
  <p>
    <h1>Página personal de Nacho Marín.</h1><br>
    <a href="biografia.html">Mi biografía</a>
    <a href="aficiones.html">Mis aficiones</a>
  </p>
  <p id="texto1">
    La multimedia tiene un montón de <a href="#aplicaciones">aplicaciones</a>
    <br> . . .
  </p>
  

  <p id="texto2">
    Entre las aplicaciones de la multimedia nos encontramos con:
    <a name="aplicaciones">
      <ul>
        <li>Sistemas educativos.</li>
        <li>Presentaciones interactivas.</li>
        <li>etc.</li>
      </ul>
    </p>
</body>
</html>
```

index.html

ancla origen



ancla destino



Concepto: URI



Uniform Resource Identifier

Cadena de caracteres con formato concreto que identifica los recursos disponibles en la Web

Formato de un URI (formal)

Véase: <http://w3.org/Addressing/>

¿URL ó URI?

El término URL está obsoleto. Utilizar URI

URL (Uniform Resource Locator) → localización de recurso

Otros tipos de URIs menos conocidos son:

URN (Universal Resource Name) → identificación única

URC (Universal Resource Characteristics) → meta-información



Concepto: URI

Ejemplos

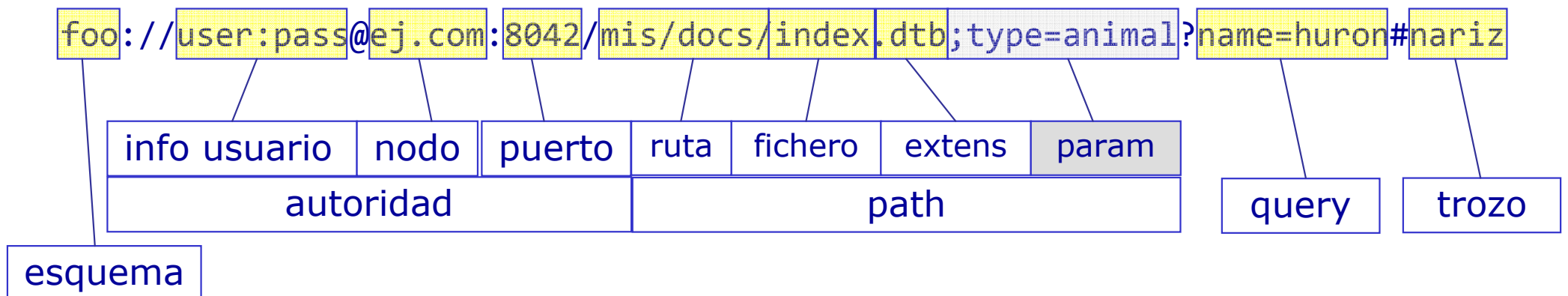
`http://www.atc.uniovi.es`

`http://www.atc.uniovi.es/index.html`

`ftp://manu:asdf1jh@156.23.34.56:3477/docs/acta.pdf`

`http://www.dyns.net/ps.php?username=joe&password=foo&host=thx`

Formato genérico (informal)





URIs relativas

- Un ancla puede referenciar un URI incompleto, al que falten los campos iniciales
- Los campos que faltan se rellenan con los mismos valores del URI del documento que lo contiene

Ejemplos.

Si el URI del documento es `http://ej.com/mis/docs/text?q`

El ancla	Se resuelve como
<code>prueba</code>	<code>http://ej.com/mis/docs/prueba</code>
<code>/prueba</code>	<code>http://ej.com/prueba</code>
<code>//otro.com</code>	<code>http://otro.com</code>
<code>http://otro.com</code>	<code>http://otro.com</code>
<code>?y</code>	<code>http://ej.com/mis/docs/text?y</code>
<code>#parte</code>	<code>http://ej.com/mis/docs/text?q#parte</code>
<code>..</code>	<code>http://ej.com/mis/</code>
<code>../prueba</code>	<code>http://ej.com/mis/prueba</code>
<code>../../prueba</code>	<code>http://ej.com/prueba</code>

Protocolo HTTP



Servidor



Cliente



```
GET http://ej.com/principal.html HTTP/1.0  
User-Agent: Mozilla/5.0 (Macintosh; ...)
```

```
HTTP/1.0 200 OK  
Date: Fri, 31 Dec 2009 14:34:00 GMT  
Content-Type: text/html  
Content-Length: 715  
<html>  
...
```



Content-Type?

- También llamados MIME-types o Content-Type
- Es una cadena ASCII estandarizada por IANA

Ejemplos:

```
"audio/mpeg"  
"text/plain"  
"application/msword"  
"image/jpeg"  
"video/mp4"  
"text/html; charset=UTF-8"  
"application/x-shockwave-flash"
```

Tipo

Subtipo

Parámetro (opcional)

x- indica no estándar



Más ejemplos

`text/html`

texto con marcas HTML

`text/plain`

texto "plano"

`image/jpg`

imagen en formato JPEG

`image/png`

imagen en formato PNG

`video/mpeg`

video en formato MPEG

`application/pdf`

documento en formato PDF

`application/zip`

archivo en formato zip

`application/octet-stream` "cualquier cosa"

`application/vnd.ms-powerpoint` presentación PowerPoint

Limitaciones de HTML



HTML no es adecuado para
Multimedia Interactiva

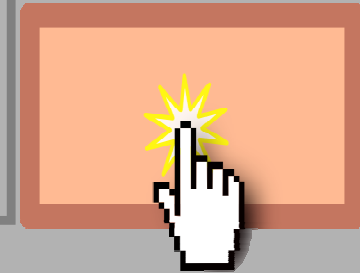
No puede
incluir
gráficos
vectoriales



No puede incluir
animaciones
(salvo GIF)



No puede incluir
elementos multimedia
sincronizados entre sí



Interactividad
muy limitada
(sólo enlaces y
áreas)

Pero hay soluciones



¡Hay otros lenguajes de marcado!



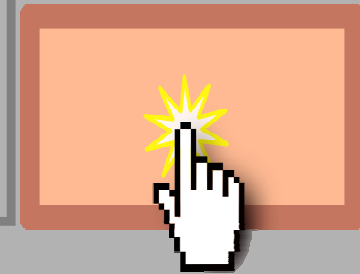
SVG



SVG+SMIL



SMIL



Lo anterior
+
Javascript

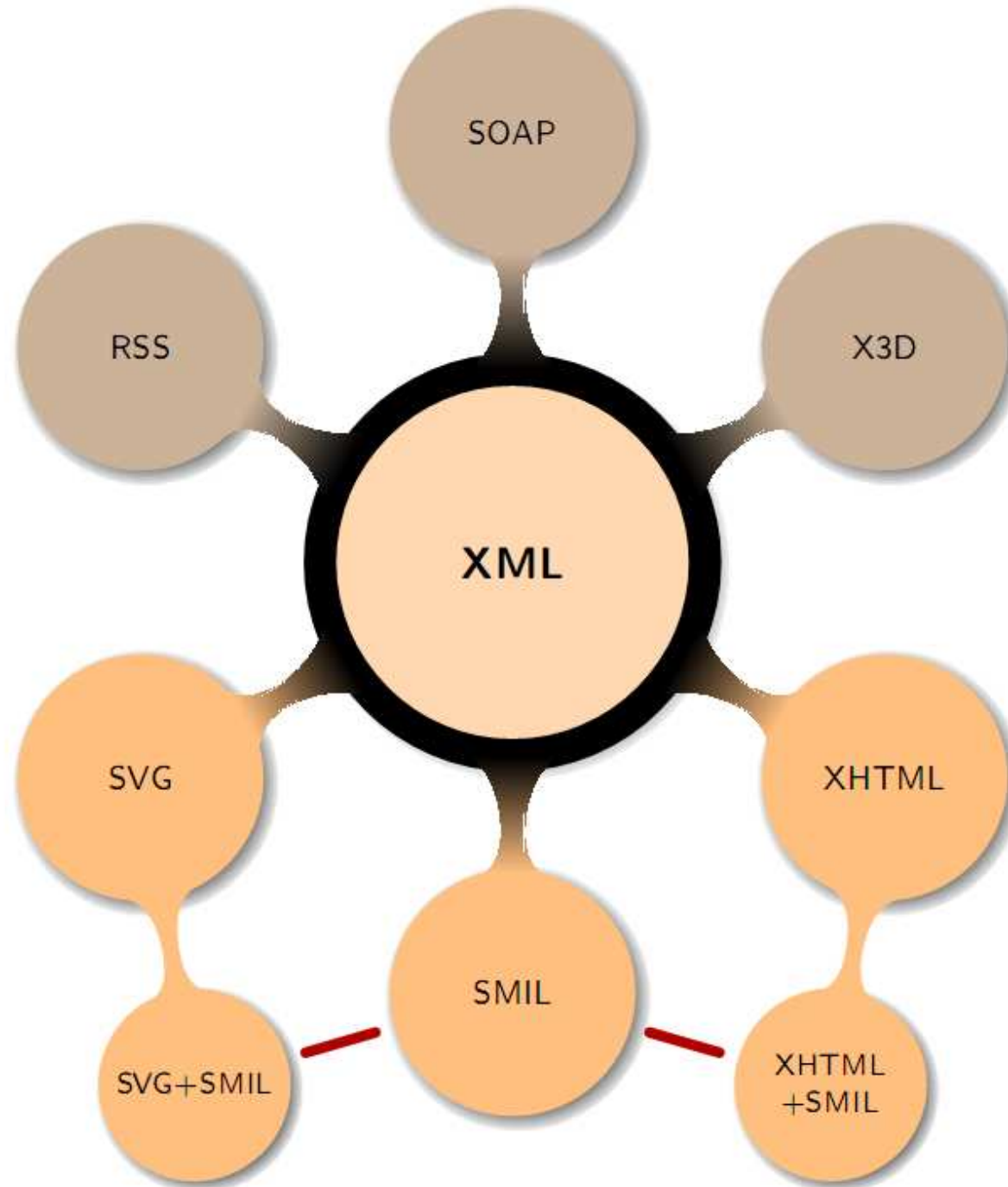
Lenguajes de marcado genéricos





- e**X**tensible **M**arkup **L**anguage
- No es un lenguaje, sino un *meta*-lenguaje
- No especifica un lenguaje de marcado, sino unas reglas para generar nuevos lenguajes de marcado
- Las reglas son simples y permiten comprobar si el documento está *bien formado*, incluso sin comprender lo que *significa*

XML se usa para crear otros lenguajes



XML: reglas sencillas



- La codificación de caracteres es Unicode
- Las etiquetas son `<asi> </asi>`
- Una etiqueta sin contenido es `<asi/>`
- SENSIBLE AL CASO
`<Asi>` no es lo mismo que `<asi>`
- Cada etiqueta (elemento) puede tener atributos y sus valores deben ir entre comillas
- Todo lo comprendido entre las etiquetas es el contenido del elemento



Documento bien formado:

- Codificación Unicode válida
- Los caracteres `<`, `>`, `&`, `"` y `'` están reservados (se representan como `<`, `>`, `&`, `"` y `'` respectivamente)
- Las etiquetas deben estar correctamente anidadas, no pueden faltar las de cierre
- Hay un único elemento raíz

¿Entonces esto es XML?

```
<html>
  <head>
    <title>Mi página personal</title>
  </head>
<body>
  <p>
    <h1>Página personal de Nacho Marín</h1><br>
    <a href="biografia.html">Mi biografía</a>
    <a href="aficiones.html">Mis aficiones</a>
  </p>
</body>
</html>
```

- HTML podría parecer un caso particular de XML
- Pero en HTML se admiten documentos *mal formados* y en XML no
- Este ejemplo no está *bien formado* ¿ves por qué?

XML Válido



- Un documento bien formado sólo verifica que los tags de apertura se cierran más adelante y la anidación es correcta
- No dice nada sobre qué nombres de elementos son válidos
 - ✓ ¿Son válidos los elementos `<title>`, `<layout>`, `<ejemplo>`, `<asdfgasdfg>`?
- Ni qué elementos pueden aparecer dentro de otros
 - ✓ ¿Puede aparecer `<body>` dentro de `<head>`?



¿Es válido este documento?

```
<lista_de_películas>
<película>
  <título>Buscando a Nemo</título>
  <audiencia>Infantil</audiencia>
  <comentario>Pezqueñines no, gracias</comentario>
</película>
<película>
  <título>Rambo</título>
  <audiencia>Todos los públicos</audiencia>
  <comentario>Tiros sin parar</comentario>
</película>
</lista_de_películas>
```

Está bien formado, pero no podemos decir más



XML Válido

- La especificación gramatical de qué nombres de elementos son válidos y en qué orden deben aparecer se especifica en un fichero DTD (*Document Type Definition*)

- Un XML válido es:

- ✓ un XML bien formado
- ✓ que especifica un DTD (o una referencia a uno)
- ✓ y que cumple con lo especificado en ese DTD





Ejemplo de DTD (peLiXML.dtd)

```
<?xml version="1.0" encoding="utf-8" ?>
  <!ELEMENT lista_de_peliculas (pelicula)*>
  <!ELEMENT pelicula          (titulo, audiencia, comentario?)>
  <!ELEMENT titulo            (#PCDATA)>
  <!ELEMENT audiencia         (#PCDATA)>
  <!ELEMENT comentario        (#PCDATA)>
```

- Mediante un meta-lenguaje especifica
 - ✓ Que el elemento raíz debe ser **<lista_de_peliculas>**
 - ✓ Que dentro de este elemento puede haber cero o más **<pelicula>**
 - ✓ Que una **<película>** tiene un **<titulo>**, una **<audiencia>** y uno o más **<comentario>**
 - ✓ Y que cada uno de estos tiene un contenido de datos

XML Válido



Ahora sí podemos responder
¿Es válido este documento?

```
<!DOCTYPE lista_de_peliculas SYSTEM "peliXML.dtd">
<lista_de_peliculas>
  <pelicula>
    <titulo>Buscando a Nemo</titulo>
    <audiencia>Infantil</audiencia>
    <comentario>Pezqueñines no, gracias</comentario>
  </pelicula>
  <pelicula>
    <titulo>Rambo</titulo>
    <audiencia>Todos los públicos</audiencia>
    <comentario>Tiros sin parar</comentario>
  </pelicula>
</lista_de_peliculas>
```

Observa
cómo incluye
el DTD

Otros esquemas de validación



- DTD es un lenguaje de "esquema" pero no el único
- Otro muy utilizado es XML Schema
 - ✓ A diferencia de DTD, un documento XML Schema es XML válido (→ hay un DTD para XML Schema)
 - ✓ ¡Incluso hay un XML Schema para XML Schema!
 - ✓ A diferencia de DTD, XML Schema puede especificar el tipo de cada contenido

Ejemplo con XML Schema



```
<lista_de_peliculas
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
  xsi:noNamespaceSchemaLocation="peliXML.xsd" >
```

```
<pelicula>
```

```
  <titulo>Buscando a Nemo</titulo>
```

```
  <audiencia>Infantil</audiencia>
```

```
  <comentario>Pezqueñines no, gracias</comentario>
```

```
</pelicula>
```

```
<pelicula>
```

```
  <titulo>Rambo</titulo>
```

```
  <audiencia>Todos los públicos</audiencia>
```

```
  <comentario>Tiros sin parar</comentario>
```

```
</pelicula>
```

```
</lista_de_peliculas>
```

Observa
cómo incluye
el esquema

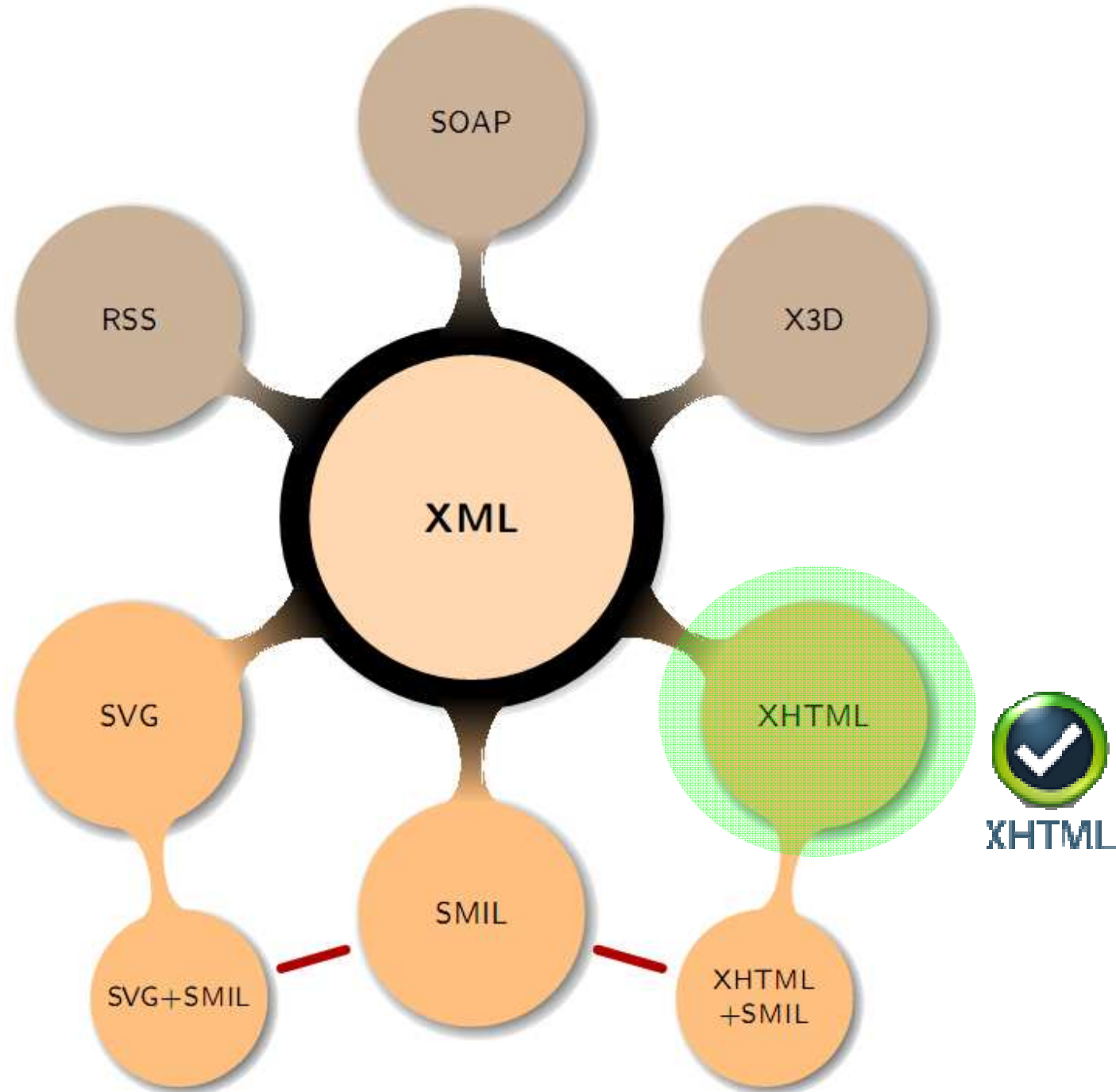


Ejemplo con XML Schema

Contenido de peliXML.xsd

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="lista_de_peliculas">
  <xs:complexType>
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:element name="pelicula">
        <xs:complexType>
          <xs:sequence>
            <xs:element name="titulo" type="xs:string"/>
            <xs:element name="audiencia" type="xs:string"/>
            <xs:element name="comentario" type="xs:string"
              maxOccurs="unbounded"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```

Lenguajes derivados de XML





- Nueva definición de HTML a partir de XML
 - ✓ Redefinición de DTDs de HTML versión 4 mediante XHTML
 - ✓ Idea: Permitir que los documentos XHTML se vean en visores de XML
- Además, ahora los documentos han de estar bien formados para visualizarlos
 - ✓ Validación del documento contra el DTD

- Diferencias entre HTML y XHTML
 - ✓ El orden de apertura de un elemento debe corresponder con el orden de cerrado del mismo
 - ✓ Las etiquetas siempre deben estar escritas en minúsculas
 - ✓ Todos los elementos de XHTML deben cerrarse, incluidos aquellos que sean una única etiqueta
 - ✓ Atributos **siempre** entre comillas



Tipos de documentos XHTML

Estrictos: cumplen de forma estricta la DTD de XHTML y no usa ninguna característica de presentación

De transición: utilizan las características de presentación de HTML para ser visualizados en navegadores que no pueden utilizar características de estilo

Para uso de marcos: dividen la ventana del navegador en varias áreas con el uso de la etiqueta FRAMESET de HTML

Indicación de tipos de documentos XHTML

Estrictos:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

De transición:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

Para uso de marcos:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
```

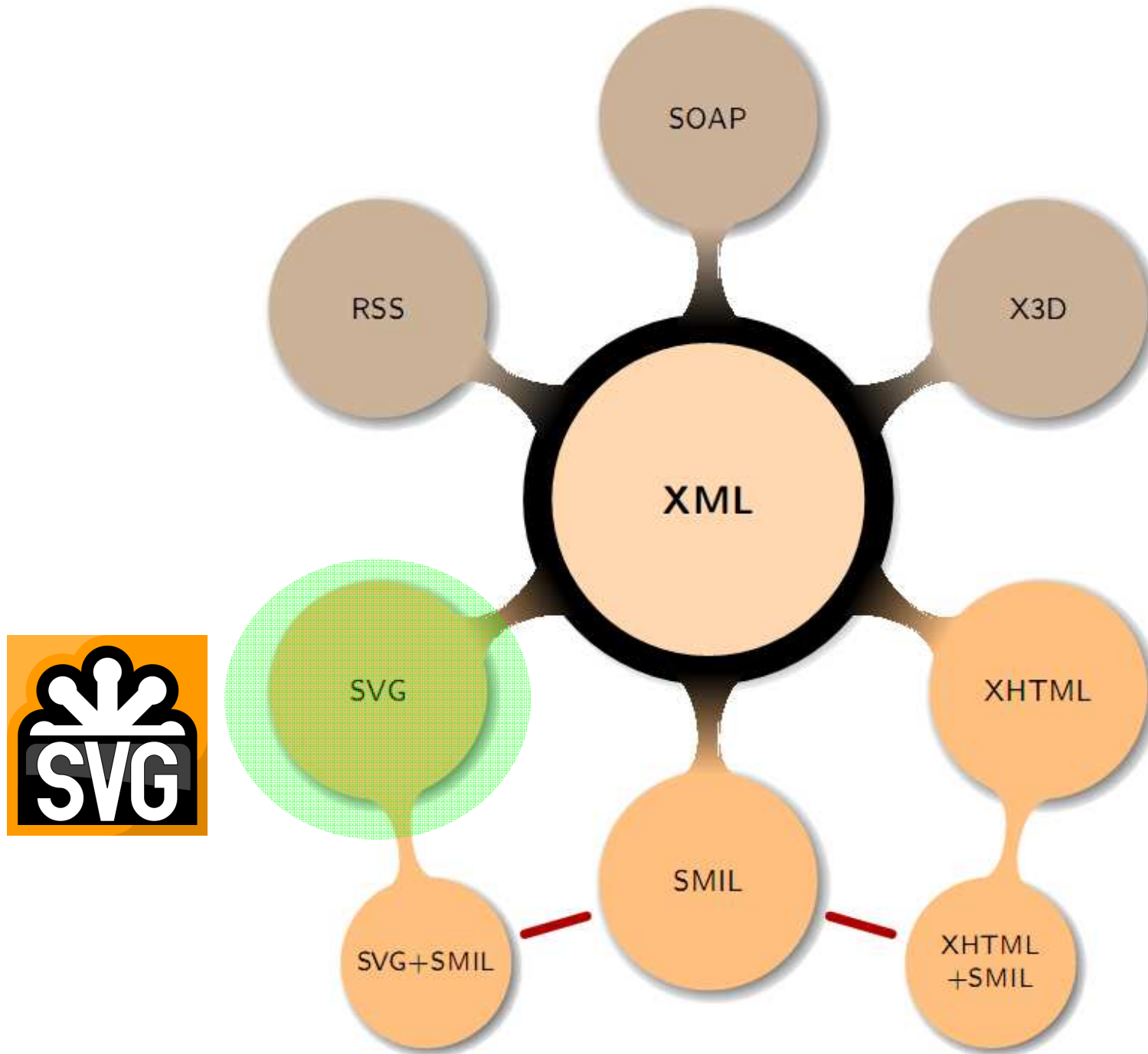
Ejemplo de XHTML estricto



XHTML

```
<!DOCTYPE html PUBLIC
  "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd" >
<html>
  <head>
    <title>Mi primera página</title>
  </head>
  <body>
    <p>
      Este es un ejemplo de página web. Voy a añadir una imagen:<br />
      
    </p>
    <p>
      <a href="http://validator.w3.org/check/referer">
        
      </a>
    </p>
  </body>
</html>
```

Lenguajes derivados de XML





- *Scalable Vector Graphics*
- Plataforma para gráficos 2D
 - ✓ Formato de fichero basado en XML
 - ✓ API de programación de aplicaciones gráficas
- Algunas características
 - ✓ Soporte de formas (shapes)
 - ✓ Texto
 - ✓ Gráficos "raster" incrustados
 - ✓ Scripting (ECMAScript)
 - ✓ Animación (SVG+SMIL)



Referencias sobre SVG

SVG Examples:

http://www.w3schools.com/svg/svg_examples.asp

SVG Reference:

http://www.w3schools.com/svg/svg_reference.asp

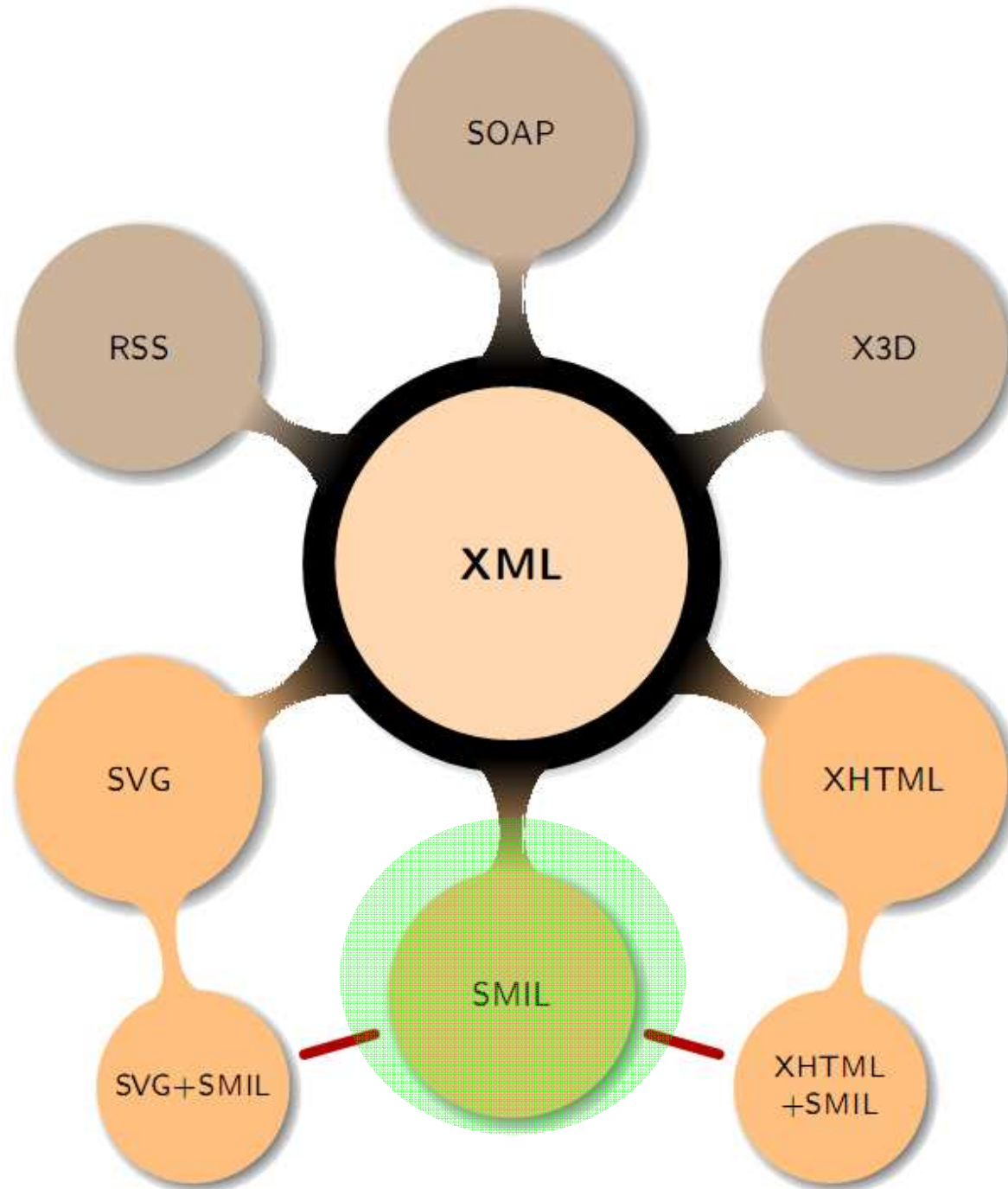
SVG a partir de XML mediante XSLT:

<http://www.carto.net/papers/svg/samples/xslt/>

Ejemplo del mapa de aeropuertos con SVG:

http://www.carto.net/papers/svg/us_weather/

Lenguajes derivados de XML



Lección 7

Recordemos...



El mercado electrónico define dos estructuras

XHTML, XML...

Estructura lógica: formada por las partes que lo componen y sus relaciones

Estructura física: indica la apariencia del documento, incluyendo sus componentes físicos, posicionamiento de elementos y tipografía empleada

Lenguajes de Presentación



- **CSS (*Cascading Style Sheets*):**
 - ✓ Define la apariencia de elementos HTML / XHTML / XML que se muestran en un navegador web
 - ✓ Permite asociar un estilo (fuentes, espaciado, colores, etc.) a un documento estructurado (HTML, XML, ...)
 - ✓ En desarrollo la versión CSS3
- **XSL (*eXtensible Stylesheet Language*):**
 - ✓ Define la presentación de documentos XML
 - ✓ El XML es transformado según reglas expresadas en lenguaje XSLT
 - ✓ El resultado se visualiza según otras reglas