UD1. INTRODUCCIÓN A LOS LENGUAJES DE MARCAS

ÍNDICE

1. OBJETIVOS DIDÁCTICOS 2
2. INTRODUCCIÓN HISTÓRICA 3
2.1. Aparición 3
2.2. Charles Goldfarb y el GML 3
2.3. TeX y LaTeX 4
2.4. RTF 5
2.5. SGML 6
2.6. Posdata 7
2.7. HTML 8
2.8. XML 8
2.9. Descuento 9
2.10. JSON 9
3. TIPOS DE LENGUAJES DE MARCAS 10
4. CARACTERÍSTICAS Y COMPONENTES 10
4.1. Etiquetas 10
4.2. Jerarquía 11
4.3. Atributos 11
4.4. Estándar 12
4.5. Facturación 12
5. HERRAMIENTAS DE EDICIÓN 12

1. OBJETIVOS DIDÁCTICOS

Los objetivos de esta Unidad Didáctica son:

- Conocer la evolución de los lenguajes de marcas desde su aparición para dar contexto al estado del arte de los actuales lenguajes.
- Identificar rápidamente los diferentes lenguajes de marcas visualizando su apariencia sin renderizar.
- Identificar las características y ámbitos de aplicación de los lenguajes de marcas más comunes.
- Conocer los componentes básicos de los lenguajes de marcas.
- Identificar herramientas de edición de lenguajes de marcas.

2. INTRODUCCIÓN HISTÓRICA

2.1. Aparición

El problema de la exportación de datos ha puesto en duda a los archivos binarios como fuente para exportar e importar información.

En su lugar, parece que los archivos de texto poseen menos problemas. Por ello, se ha intentado que los archivos de texto plano (archivos que sólo contienen texto y no otros datos binarias) pudieran servir para almacenar otros datos como por ejemplo detalles sobre el formato del propio texto u otras indicaciones.

Los procesadores de texto fueron el primer software en encontrarse con este dilema. Al ser programas que sirven para escribir texto parecía que lo lógico era que sus datos se almacenaran como tal. Pero necesitan guardar datos referidos al formato del texto, tamaño de la página, color, tipografía, márgenes, etc. La solución clásica ha sido guardar la información de manera binaria, lo que provoca problemas.

Algunos procesadores de texto optaron por guardar toda la información como texto, haciendo que las indicaciones de formato no se almacenan de manera binaria sino textual. Estas indicaciones son caracteres marcados de manera especial para que así un programa adecuado pueda traducir estos caracteres no como texto sino como operaciones que finalmente producirán mostrar el texto del documento de manera adecuada.

La idea del marcado procede del inglés marking up término con el que se referían a la técnica de marcar manuscritos con lápiz de color para hacer anotaciones como por ejemplo la tipografía a emplear en las imprentas. Este mismo término se ha utilizado para los documentos de texto que contienen comandos o anotaciones.

Las posibles anotaciones o indicaciones incluidos en los documentos de texto han dado lugar a lenguajes (entendiendo que en realidad son formatos de documento y no lenguajes en el sentido de los lenguajes de programación de aplicaciones) llamados lenguajes de marcas, lenguajes de marcado o lenguajes de etiquetas.

2.2. Charles Goldfarb y el GML

Se considera a Charles Goldfarb como el padre de los lenguajes de marcas. La razón para esta consideración es, precisamente, su ayuda en la creación del lenguaje GML (Generalized Markup Language).

Golfarb era un investigador de IBM que propuso ideas para que los documentos de texto que incluyeran la posibilidad de marcar el formato del mismo. Al final ayudó a realizar el lenguaje GML de IBM el cual puso los cimientos del futuro SGML (padre de HTML y XML) ideado por el propio Goldfarb y padre de la mayoría de lenguajes de marcas actuales.

```
:h0.El reino de los animales
:h1.Mamíferos
:p.Los mamíferos (:hp1.Mammalia:ehp1.) son una clase de
vertebrados :hp2.amniotas homeotermos:ehp2. que poseen
glándulas mamarias productoras de leche con las que
alimentan a las crías
:h1.Aves
:p. Las aves son animales vertebrados, de sangre
caliente, que caminan, saltan o se mantienen solo sobre
las extremidades posteriores
```

Este código renderizado por un software que interprete este código obtendría como resultado:

El reino de los animales

Mamíferos

Los mamíferos (**Mammalia**) son una clase de vertebrados ***amniotas homeotermos***, que poseen glándulas mamarias productoras de leche con las que alimentan a las crías

Aves

Las aves son animales vertebrados, de sangre caliente, que caminan, saltan o se mantienen solo sobre las extremidades posteriores

2.3. TeX y LaTeX

En la década de los 70 Donald Knuth (uno de los ingenieros informáticos más importantes de la historia, padre del análisis de algoritmos y premio Turing 1974) creó el lenguaje TeX para producir documentos científicos utilizando una tipografía y capacidades que fueron iguales en cualquier computadora, asegurando además una gran calidad en los resultados.

Para ello hizo alusión a TeX con tipografía especial (fuentes Modern Computer) y un lenguaje de definición de tipo (METAFONT). TeX ha tenido un cierto éxito en la comunidad científica gracias a sus 300 comandos que permiten crear documentos con tipos de gran calidad. Requiere de software capaz de convertir el archivo TeX a un formato de impresión.

El éxito de TeX produjo numerosos derivados de los cuales el más popular es LaTeX. LaTeX fue definido en 1984 por Leslie Lamport (premio Turing 2003), aunque luego ha sido numerosas veces revisado. Al utilizar comandos de TeX y toda su estructura tipográfica, adquirió rápidamente notoriedad y sigue siendo utilizado para producir documentos con expresiones científicas, de gran calidad. La idea es que los científicos se centren en el contenido y no en la presentación.

```
\documentclass[12pt]{article}
\usepackage{amsmath}
\title{\Ejemplo}
\begin{document}
Este es el texto ejemplo de \LaTeX{}
Con datos en \emph{cursiva} o \textbf{negrita}.
Ejemplo de f\'ormula
\begin{align}
E &= mc^2
\end{align}
\end{document}
```

Que con un traductor daría lugar al resultado:

Este es el texto ejemplo de LaTeX Con datos en *cursiva* o **negrita**. Ejemplo de formula

$$E = mc^2 \tag{1}$$

2.4. RTF

RTF es el acrónimo de Rich Text Format (Formato de Texto Enriquecido) un lenguaje ideado por Richard Brodie, Charles Simonyi y David Luebbert (miembros del equipo de desarrollo de Microsoft Word) en 1987 para producir documentos de texto que incluyan anotaciones del formato. Es un formato propiedad de Microsoft, pero reconocido por la mayoría de aplicaciones de proceso de texto.

Actualmente se trata de un formato aceptado para documentos de texto que cuenten con información sobre el estilo del texto. Se usa mucho como formato de intercambio entre diferentes procesadores por su potencia. El procesador de texto Word Pad incorporado dentro del sistema operativo Windows lo utiliza como formato nativo.

Codifica el texto usando 8 bits, para caracteres fuera de la ASCII requiere de secuencias de escape lo que, prácticamente, le hace inviable como formato legible de texto en la mayoría de lenguas del planeta. En las últimas versiones de RTF ya sí se ofrece un mayor apoyo a Unicode.

Su éxito procede a que las indicaciones de formato son potentes y son más legibles para las personas que el formato nativo de los procesadores de textos, aunque es, como lenguaje de marcado, uno de los más crípticos.

```
{\rtf\ansicpg1252\deff0\deflang3082
{\fonttbl
{\f0\fcharset0\froman Times New Roman}
{\f1\fcharset0\fswiss Arial Black}
}
{\pard \f1\fs48
El reino de los animales
\par}
{\pard \f1\fs40
Mamíferos
\par}
{\pard \f0\fs25
Los mamíferos ({\b Mammalia}) son una clase de
vertebrados {\i amniotas homeotermos} que poseen
glándulas mamarias productoras de leche con las que
alimentan a las crías
\par}
{\pard \f1\fs40
Aves
\par}
{\pard \f0\fs25
Las aves son animales vertebrados, de sangre caliente,
que caminan, saltan o se mantienen solo sobre las
extremidades posteriores
\par}
}
```

Produce el resultado:

2.5.SGML

Se trata de una mejora muy notable del lenguaje de GML que estandarizaba el lenguaje de marcado y que fue definida finalmente por ISO como estándar mundial en documentos de texto con etiquetas de marcado. Su responsable fue Charles Goldfarb.

Su importancia radica en que es el padre del lenguaje XML y la base sobre la que se sostiene el lenguaje HTML, dos de los lenguajes de marcas más populares de la historia.

En SGML los elementos que contienen indicaciones para el texto se colocan entre símbolos < y >. Las etiquetas se cierran con el signo /. Es decir las reglas fundamentales de los lenguajes de etiquetas actuales ya las había definido SGML.

En realidad (como XML) no es un lenguaje con unas etiquetas concretas, sino que se trata de un lenguaje que sirve para definir lenguajes. Entre los lenguajes definidos mediante SGML, sin duda HTML es el más popular.

```
<articulo>
        <titulo1>El reino de los animales</titulo1>
        <titulo2>Mamíferos</titulo2>
        <normal>Los mamíferos (<negrita>Mammalia</negrita>)
        <titulo2>Aves</titulo2>
        <normal>Las aves son animales vertebrados, de sangre
</articulo>
```

Como veremos más adelante, este documento es muy parecido a un documento realizado en XML, de hecho XML es un subconjunto de SGML más restrictivo (es un lenguaje que tiene normas más estrictas).

SGML necesitará definir cómo se debe mostrar los elementos titulo1, titulo2, etc. Ya que son nombres de elementos que habrá que definir. Esa es la prueba de que es un lenguaje para definir tipos de documento.

SGML aportó las etiquetas tal cual las conocemos actualmente gracias al éxito de HTML.

2.6. Posdata

Se trata de un lenguaje de descripción de páginas. De hecho es el más popular para este fin, siendo el lenguaje más utilizado por los sistemas de impresión de alta gama.

Permite crear documentos en los que se dan indicaciones potentísimas sobre cómo mostrar información en el dispositivo final. Se inició su desarrollo en 1976 por John Warnock y dos años más tarde se continuó con la empresa Xerox, hasta que en 1985 el propio Warnock crea Adobe Systems y desde esa empresa se continua su desarrollo.

Es en realidad todo un lenguaje de programación que indica la forma en que se debe mostrar la información que puede incluir texto y el tipo de letra del mismo, píxeles individuales y formas vectoriales (líneas, curvas). Sus posibilidades son muy amplias.

```
%colocar el cursor
100 100 moveto
%dibuja cuadrado
100 200 lineto
200 200 lineto
200 100 lineto
100 100 lineto
%relleno
stroke
```

2.7. HTML

Tim Bernes Lee utilizó SGML para definir un nuevo lenguaje de etiquetas que llamó Hypertext Markup Language (lenguaje de marcado de hipertexto) para crear documentos transportables a través de Internet en los que fuera posible el hipertexto; es decir la posibilidad de que determinadas palabras marcadas de manera especial permitieran abrir un documento relacionado con ellas.

A pesar de tardar en ser aceptado, HTML fue un éxito rotundo y la causa indudable del éxito de Internet. Hoy en día casi todo en Internet se ve a través de documentos HTML, que popularmente se denominan páginas web.

Inicialmente estos documentos se veían con ayuda de intérpretes de texto (como por ejemplo el Lynx de Unix) que simplemente colorearían el texto y remarcaban el hipertexto. Después el software se mejoró y aparecieron navegadores con capacidad más gráfica para mostrar formatos más avanzados y visuales.

Ejemplo (usando el mismo contenido de los ejemplos anteriores):

```
<!DOCTYPE html>
<html lang="es">
<head>
        <meta charset="UTF-8">
        <title>Document</title>
</head>
<body>
        <h1>El reino de los animales</h1>
        <h2>Mamíferos</h2>
        <p>Los mamíferos (<strong>Mammalia</strong>) son una clase de vertebrados
        <h2>Aves</h2>
        <p>Las aves son animales vertebrados, de sangre caliente, que caminan, sa
</body>
</html>
```

2.8. XML

Se trata de un subconjunto de SGML ideado para mejorar el propio SGML y con él definir lenguajes de marcado con sintaxis más estricta, pero más comprensible.

Ha sido enormemente popular desde finales de los 90 y ha conseguido incorporar numerosos lenguajes a su alrededor para conseguir documentos muy dinámicos y con gran capacidad de formato. Es uno de los formatos de documentos más populares para exportación e importación de datos.

Actualmente está siendo sobrepasado en la mayoría de sus usos por JSON

```
<?xml version="1.0" encoding="UTF-8"?>
<nombre>Jorge</nombre>
<apellido1>Sánchez</apellido1>
<dirección>
  <calle>C/ Falsa nº 0</calle>
  <localidad>Palencia</localidad>
  <código_Postal>34001</código_Postal>
  <pais>España</pais>
</direccion>
<teléfonos>
  <teléfono tipo="fijo">999 999 999</teléfono>
  <teléfono tipo="móvil">666 666 666</teléfono>
</telefono>
```

2.9. Descuento

Se trata de un formato de marcado simple que permite crear documentos sencillos y convertirlos en documentos HTML.

Fue creado por John Gruber con la ayuda de Aaron Shwartz. La pretensión de este lenguaje es definir unas normas muy sencillas para crear documentos parecidos a los que se crean mediante el lenguaje HTML.

Ha tenido un éxito muy notable, especialmente desde que fue adoptado por sitios tan populares como GitHub, Reddit o StackExchange para que los usuarios publicaran contenido con formato.

Ejemplo de texto con formato Markdown:

```
# El reino de los animales
## Mamíferos
Los mamíferos (**Mammalia**) son una clase de vertebrados *amniotas homeotermos*. que poseen gl
## Aves
Las aves son animales vertebrados, de sangre caliente, que caminan, saltan o se mantienen solo
```

2.10. JSON

Abreviatura de JavaScript Object Notation, Se trata de una notación de datos procedente del lenguaje JavaScript estándar (concretamente en la versión ECMAScript de 1999). En el año 2002 se le apoyaba desde muchos de los navegadores y su fama ha sido tal que ahora se ha convertido en una notación independiente de JavaScript que compite claramente con XML en funcionalidad.

Las razones de su éxito se deben a su versatilidad, ya que permiten definir datos complejos, como arrays o código de funciones, elementos pertenecientes al mundo de la programación de aplicaciones. El éxito de JavaScript junto a la versatilidad comentada, le han convertido en el lenguaje de marcado más popular para almacenar datos.

En JSON, el texto se divide en datos y metadatos. De manera que el símbolo de los dos puntos separa el metadato del dato. Por otro lado, los símbolos de clave y claudátor permiten agrupar de diversas formas los datos.

Ejemplo de código JSON:

```
{
    "nombre": "Jorge",
    "apellido1": "Sánchez",
    "dirección": {
        "calle": "C/ Falsa nº 0",
        "localidad": "Palencia",
        "código Postal": 34001,
        "país":"España"
    },
    "teléfonos": [
        {
            "tipo": "fijo",
            "número": "999 999 999"
        },
        {
            "tipo": "móvil",
            "number": "666 666 666"
        }
    ]
}
```

3. TIPOS DE LENGUAJES DE MARCAS

Se pueden clasificar en:

- **Orientados a la presentación.** En ellos los metadatos permiten indicar el formato en el que se debe presentar el texto. Es el caso de RTF, en el que sus etiquetas especifican tipos de letra, tamaños de página, colores, etc. Las primeras versiones de HTML también se consideran así, ya que incluían etiquetas como fuente mediante la cual se especificaba el formato de fuente.
- **Orientados a la descripción.** En ellos las marcas especiales permiten dar significado al texto pero no indican cómo se debe presentar en pantalla lo mismo. Sería el caso de XML (o de SGML), JSON, Markdown y de las versiones actuales de HTML. En estos lenguajes simplemente se indica el significado del contenido: si el texto es un título, un párrafo normal, un pie de ilustración, una dirección postal etc.
- **Orientados a procedimientos.** Se trata de documentos en los que el texto marcado, se interpreta como órdenes a seguir, y así el archivo en realidad contiene instrucciones a realizar con el texto (girarle, convertirle en una fórmula, realizar una suma, etc.). Es el caso de LaTeX o PostScript.

4. CARACTERÍSTICAS Y COMPONENTES

4.1. Etiquetas

Los lenguajes de marcas utilizan una serie de etiquetas especiales intercaladas en un documento de texto sin formato. Estas etiquetas serán posteriormente interpretadas por los intérpretes del lenguaje y ayudan al procesamiento del documento.

Las etiquetas se escriben cerradas entre ángulos, es decir < y >. Normalmente, se utilizan dos etiquetas: una de inicio y otra de fin para indicar que ha terminado el efecto que queríamos presentar. La única diferencia entre ambas es que la de cierre lleva una barra inclinada "/" antes del código.

Las últimas especificaciones emitidas por el W3C indican la necesidad de que vayan escritas siempre en minúsculas para considerar que el documento está correctamente creado.

4.2.Jerarquía

La posición de las marcas a este tipo de lenguajes es muy importante. Si una marca está contenida dentro de otra se dice que pertenece a su jerarquía. La jerarquía es de mucha importancia tanto en los Lenguajes de marcas orientados a la presentación como a los orientados a la descripción.

En el ejemplo del siguiente punto, todos los ejemplares de libro pertenecen a la biblioteca, y cada autor pertenece a un ejemplar en concreto.

4.3.Atributos

Permiten añadir propiedades a los elementos de un documento. Los atributos no pueden organizarse en ninguna jerarquía, no pueden contener ningún otro elemento o atributo y no reflejan ninguna estructura lógica.

No se debe utilizar un atributo para contener información susceptible de ser dividido.

Los atributos están presentes en la basta mayoría de los lenguajes de marcas modernos. A continuación podemos observar un ejemplo en XML.

```
<?xml version="1.0" encoding="iso-8859-1" standalone="yes" ?
<!DOCTYPE biblioteca >
<biblioteca>
   <ejemplar tipo_ejem="libro" titulo="XML práctico">
      <tipo> <libro isbn="978-2-7460-4958-1" edicion="1" </libro> </tipo>
      <autor nombre="Sebastien Lecomte"></autor>
      <autor nombre="Thierry Boulanger"></autor>
      <autor nombre="Angel Belinchon Calleja" funcion="traductor"></autor>
      <prestado lector="Pepito Grillo">
         <fecha_pres dia="13" mes="mar" año="2009"></fecha_pres>
         <fecha_devol dia="21" mes="jun" año="2009"></fecha_devol>
      </prestado>
   </ejemplar>
</biblioteca>
```

Como se observa en el ejemplo, los atributos se definen y dan valor dentro de una etiqueta de inicio o de elemento vacío, a continuación del nombre del elemento o de la definición de otro atributo siempre separado de ellos por un espacio. Los valores del

atributo van precedidos de un igual que sigue al nombre del mismo y deben definirse entre comillas simples o dobles.

Los nombres de los atributos deben cumplir las mismas reglas que los de los elementos, y no pueden contener el carácter menor que, <.

4.4. Estándar

Para poder transmitir la información, es fundamental hablar el mismo idioma. Para las marcas pasa lo mismo, hay que establecer un estándar para interpretar las marcas.

En los lenguajes de marcas modernos que parten de SGML como HTML y XML los estándar están establecidos por la World Wide Web Consortium (W3C).

Los documentos denominados como «bien formados» (del inglés well formed) son aquellos que cumplen con todas las definiciones básicas de formato y pueden, por tanto, analizarse correctamente por cualquier analizador sintáctico (parser) que cumpla con la norma. Se separa esto del concepto de validez que se explica más adelante.

4.5. Facturación

Que un documento esté «bien formado» solamente se refiere a su estructura sintáctica básica, es decir, que se trate de elementos, atributos y comentarios como el estándar especifica que se escriban. Ahora bien, en algunos lenguajes como XML, necesitan especificar cuál es exactamente la relación que debe verificarse entre los diferentes elementos presentes en el documento.

Esta relación entre elementos se especifica en un documento externo o definición (expresada como DTD o como XSchema). Crear una definición equivale a crear un nuevo lenguaje de marcado, para una aplicación específica.

5. HERRAMIENTAS DE TRABAJO

Las herramientas de edición de un lenguaje de marcas son normalmente los editores de texto. En la actualidad la mayor parte de editores de código tienen herramientas para hacer más fácil el trabajo con cada tipo de lenguaje. Las herramientas fundamentales son:

- Visualización de etiquetas resaltadas en el texto.
- Identificación de atributos resaltados.
- Autocompletar las etiquetas y cerrarlas automáticamente.
- Análisis sintáctico y formal del código desarrollado.
- Contracción jerárquica de etiquetas.

El herramiento de Visual Studio Code de Microsoft permite trabajar con mucho tipo de Lenguajes. Es el herramiento recomendado por esta asignatura.

→ HAZ LA PRÁCTICA 1.1←