

IV Congreso Latinoamericano de Traducción e Interpretación

**¿HABLA XML?
CURSO INTENSIVO DE LENGUAJES DE
MARCADO PARA EL TRADUCTOR
TÉCNICO**

Romina L. Marazzato

¿HABLA XML?

CURSO INTENSIVO DE LENGUAJES DE MARCADO PARA EL TRADUCTOR TÉCNICO

Romina L. Marazzato

Resumen

La explosión de nuevas tecnologías de traducción, como memorias de traducción, programas de localización y XML, ha hecho de muchos traductores espectadores boquiabiertos de una industria construida sobre sus propios hombros. Los lenguajes de marcado o LM constituyen una herramienta cada vez más difundida para el intercambio de información, especialmente el lenguaje de marcado extensible o XML. Los traductores se ven forzados a trabajar en archivos escritos en XML o utilizar programas basados en él. Esta ponencia tiene como objetivo presentarle al traductor los LM, ya que una comprensión de estas tecnologías es fundamental para trabajar con ellas. Cubriremos primero conceptos básicos sobre LM. En segundo lugar, presentaremos el lenguaje de marcado hipertextual o HTML, como punto de partida en la comprensión de otros lenguajes de marcado relevantes en traducción. Luego, pasaremos a los conceptos propios de XML. Finalmente señalaremos sus diferencias con HTML.

Introducción

Como anuncia el título de la ponencia, vamos a hablar de lenguajes de marcado o LM y su impacto en el mundo de la traducción. Me voy a dedicar específicamente a dos de los dialectos llamados lenguaje de marcado hipertextual o HTML y lenguaje de marcado extensible o XML, por sus siglas en inglés.

Comenzaremos por HTML, que es un lenguaje más simple. HTML o lenguaje de marcado hipertextual fue el primer lenguaje de marcado especialmente diseñado para la Web. Su desarrollo comenzó en los 90. Como si se tratara de inglés controlado, el HTML es un subconjunto de un lenguaje de marcado mucho más abarcativo llamado SGML o lenguaje de marcado estándar generalizado. SGML se comenzó a desarrollar en los 80 y se transformó en una norma ISO en 1986 (ISO 8879). Se emplea principalmente en la creación de documentación técnica para las industrias aeroespacial y automotriz. Básicamente, determina la naturaleza de los datos de un documento y señala su relación con el resto del documento¹.

¹ José MARTÍNEZ DE SOUSA. "Las nuevas tecnologías en el tratamiento de los textos (Los neotipógrafos)". Revista Española de Bibliología. Vol. 1, No. 1. Salamanca, mayo 1997.

Como dijimos, HTML es comparable al inglés controlado. Presenta muchas de sus características gramaticales y léxicas sin incluir sus complicaciones. El problema del inglés controlado es que sólo sirve para crear cierto tipo de documentos; ni Shakespeare ni Joyce podrían haber creado sus obras con él. Así también, el HTML ha demostrado ser insuficiente para manipular toda la información que los usuarios de HTML deseaban manipular.

Se comenzó entonces, en el año 1996, el desarrollo de otro subconjunto de SGML llamado XML o lenguaje de marcado extensible. XML es una iniciativa del World Wide Web Consortium o W3C. Se diseñó originalmente para manipular la publicación electrónica a gran escala. En febrero de 1998 se convirtió en una recomendación del W3C, es decir que es una tecnología bastante nueva. Hoy en día, XML juega un importante papel como tecnología de intercambio de información de todo tipo (no sólo técnica) y es un complemento, más que una alternativa de HTML.

El W3C es una organización que se dedica al desarrollo de tecnologías aplicadas principalmente a la Web. Para mayor información, los invito a visitar su sitio Web en www.w3.org.

¿Qué es un lenguaje de marcado?

Comencemos por el concepto de “marcado o marcación”. El concepto se deriva del mundo editorial, donde los tipógrafos necesitan distinguir entre el texto a imprimirse y las instrucciones sobre cómo imprimirlo. A ese conjunto de instrucciones, se le llama “marcado”. Por ejemplo, un marcado de título indicará que el título de un capítulo de un libro debe imprimirse en una sola línea con un tamaño de tipografía varios puntos más grande que el resto del texto, a fin de que los lectores infieran que se trata del título.

En términos generales, el marcado de un texto es una metodología para codificar datos que contengan información sobre sí mismos. Además del marcado tipográfico en el mundo editorial, la vida cotidiana está llena de ejemplos de marcado.

Un ejemplo común es el de resaltar con un fibrón libros de texto para estudiar y luego poder repasar los puntos importantes en época de exámenes. Todos en algún momento hemos usado este método o recurrido al ejemplar de algún piadoso amigo que lo utilizó antes que nosotros. Al resaltar las oraciones, párrafos o fragmentos que consideramos importantes, estamos “marcando” o codificando información. El texto ya no sólo presenta su contenido original, sino que ofrece una clasificación de dicha información, que ahora se divide en importante y no importante.

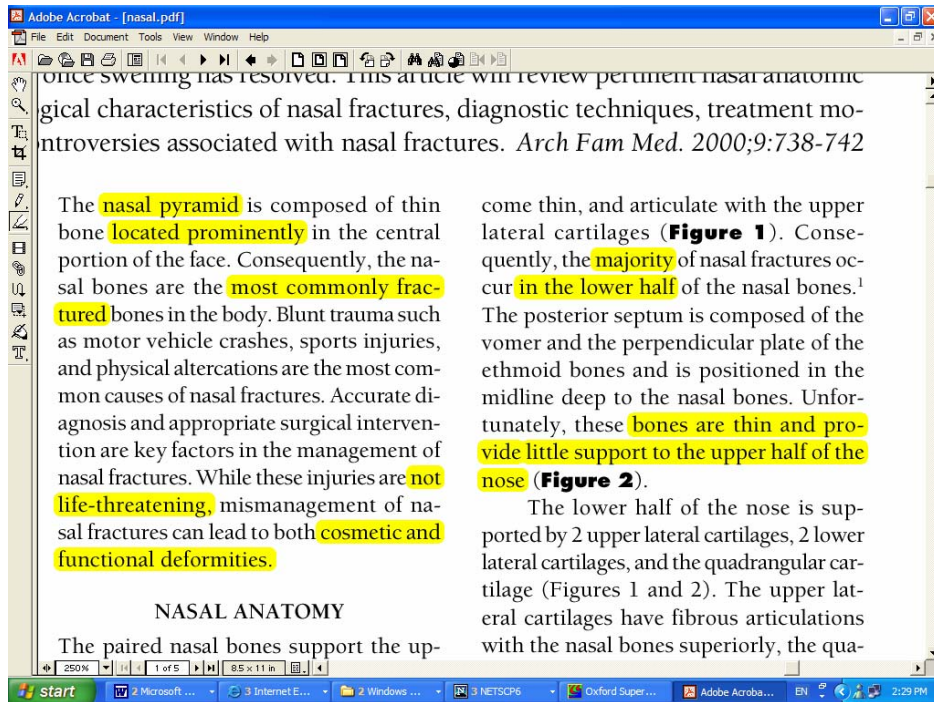


Figure 1: Marcado de libros de texto para estudiar.

Algunos estudiantes eligen incluso utilizar diversos colores para resaltar ideas principales, ideas secundarias y detalles dignos de recordar. Ahora bien, si ustedes eligen rojo para resaltar las ideas principales, verde para las ideas secundarias y azul para los detalles y yo elegí azul para las ideas principales, rojo para las ideas secundarias y verde para los detalles, no vamos a poder intercambiar nuestros textos.

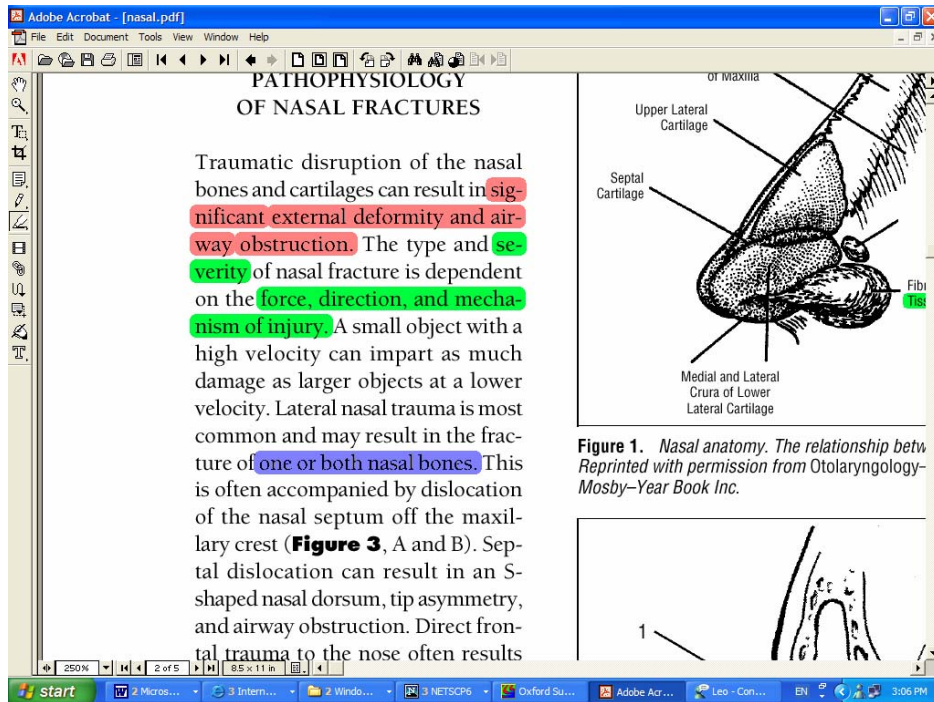


Figura 2: Marcado de libros de texto para estudiar.

De esto se desprende que un marcado será útil para un conjunto de usuarios si se cumplen dos condiciones:

1. Se debe contar con una norma que defina lo que constituye un marcado válido.
2. Se debe contar con una norma que defina lo que ese marcado significa.

En el primer ejemplo, el resaltar con un fibrón es un marcado válido, sin importar el color, ya que hay sólo un mensaje que codificar: *importante*. En el segundo ejemplo, tenemos que decidir si vamos a usar mi convención o la de ustedes. Pero cualquiera que elijamos nos permite codificar nuestros mensajes: *principal, secundario, detalle*. En términos lingüísticos, estamos definiendo una lengua mediante su sistema de diferencias², al otorgarle valores a nuestros signos (rojo, azul y verde) que se derivan de sus relaciones mutuas y excluyentes en nuestro sistema de colores.

Otro ejemplo caro para los traductores es el del sistema de signos para la corrección de textos. Muchos de nosotros aún hoy en día, en la era digital, tenemos que corregir textos en papel que deben volver a sus autores o van dirigidos a diseñadores o a la imprenta y debemos usar marcas de corrección que instruyan claramente cómo modificar el texto.

² Ferdinand DE SAUSSURE. Curso de Lingüística General. Buenos Aires. Losada. 1971. Novena edición.

Sin embargo, los tipos de marcas y sus significados no están tan normativizados como deberían y muchas veces pasamos largo rato en el teléfono explicando qué queremos indicar con ese símbolo que parece un rulito o una P al revés. Lo que estamos haciendo es definir el significado de nuestro lenguaje de marcado.

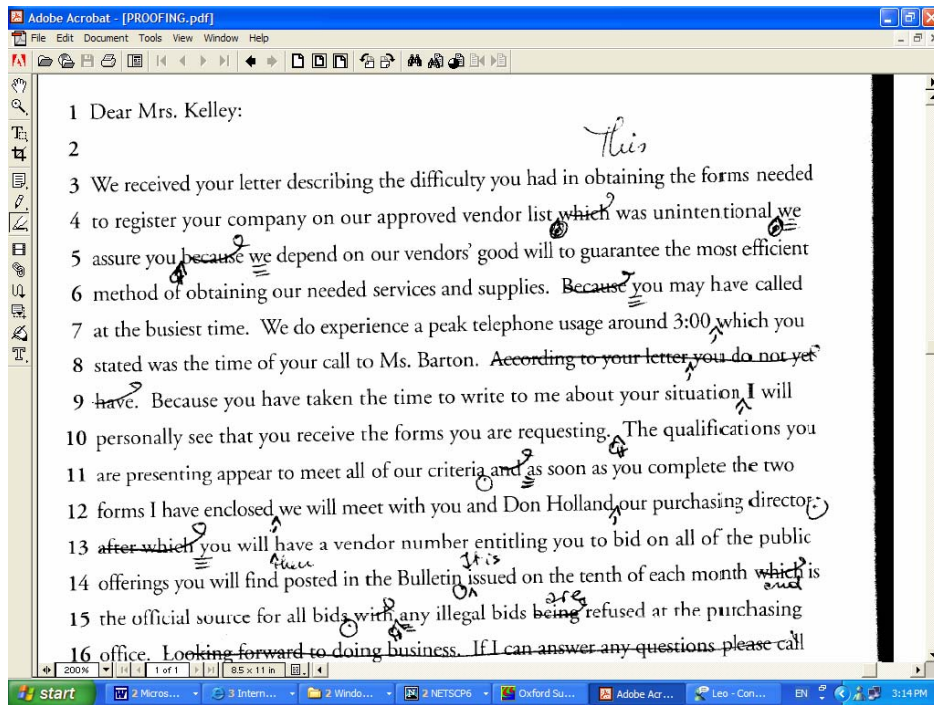


Figura 3: Marcado de corrección de textos.

A título informativo, se muestran en la siguiente tabla el significado de algunos símbolos comunes de corrección.

Símbolo de corrección	Significado
^	insertar
=	justificar horizontalmente
	justificar verticalmente
≡	usar mayúscula
/lc	usa minúscula

Tabla 1: Ejemplos de símbolos de corrección

En el mundo informático los lenguajes de marcado se utilizan en todo tipo aplicaciones. Por ejemplo, los procesadores de texto emplean el marcado para especificar el formato y la distribución del texto en la página. Cuando usamos los botones de negritas o cursiva, la justificación a la derecha o el menú de estilos en Word, como se muestra en la siguiente figura, no estamos haciendo otra cosa que indicarle al programa que marque el texto seleccionado de manera de diferenciarlo del

resto. Los botones o los menús indican el significado del marcado que va a usar el programa.

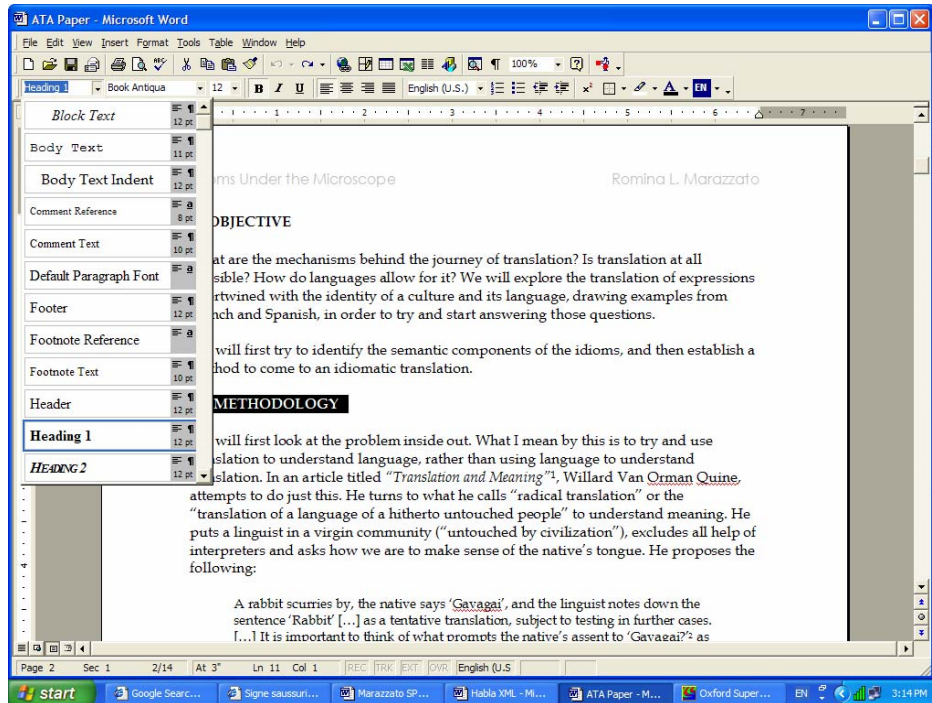


Figura 4: Marcado de diseño en un procesador de texto.

Los traductores que usan programas de traducción asistida conocen las famosas etiquetas que deben defenderse a capa y espada, so pena de destruir la magia del programa. Las etiquetas delimitan las unidades de traducción, que constan de un fragmento de texto en idioma de partida con su respectiva traducción al idioma de llegada. Pero las herramientas de traducción no sólo marcan el texto para indicar qué parte es un fragmento de partida y qué parte es un fragmento de llegada, sino que pueden incluir datos como fecha y hora de creación de la traducción, autor de la unidad de traducción, etc.

En la figura y tabla siguientes se muestran tres unidades de traducción con sus respectivas etiquetas en un procesador de texto.

Marcado en memorias de traducción	Significado
[{0}]	comienzo de la unidad
[< >]	fragmento de partida
[<189%>] [<10%>] [<196%>]	porcentaje de correspondencia en la memoria
[< >]	fragmento de llegada
[<0}]	fin de la unidad

Tabla 2: Etiquetas de marcado de herramientas de traducción

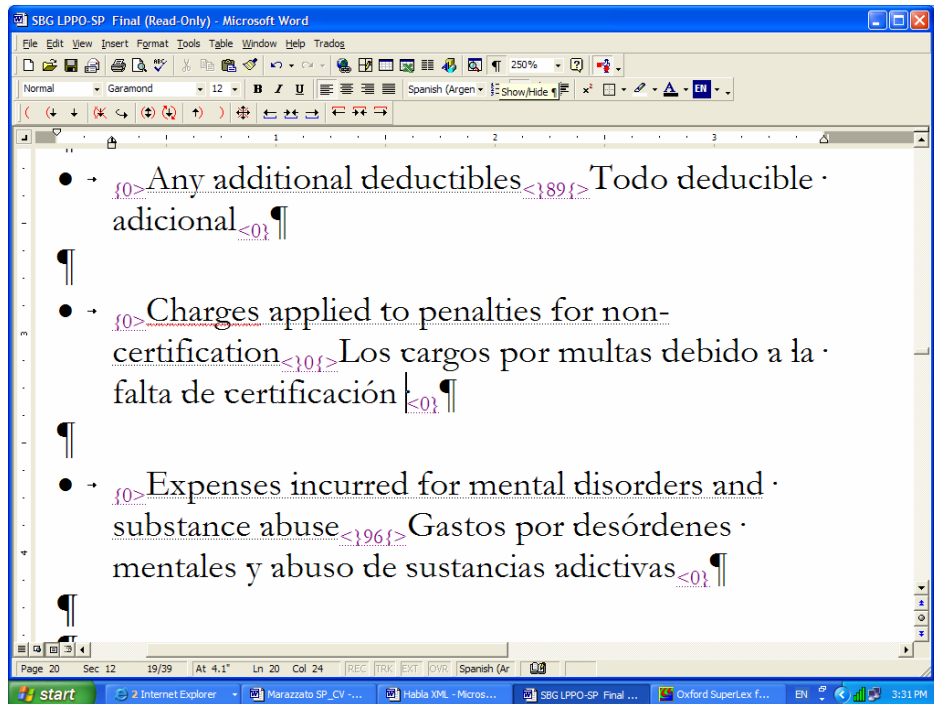


Figura 5: Marcado de en herramientas de traducción.

Todos estos ejemplos no hacen más que confirmar una realidad semiótica que todos conocemos: para hablar de un lenguaje necesitamos siempre de otro lenguaje. Como ya conocían los escolásticos, en su *teoría de las suposiciones*³, las expresiones pueden referirse a las entidades, *suppositio formalis*, o al nombre de las mismas, *suppositio materialis*, aunque carecían de una convención escrita para distinguirlas, guiándose simplemente por el contexto, cosa que las computadoras no son capaces de hacer.

Una cosa es usar el lenguaje para decir “La circunferencia es una curva cerrada cuyos puntos son equidistantes de otro, el centro, situado en el mismo plano” y otra mencionarlo para decir “‘Circunferencia’ tiene cinco sílabas.”

Esta distinción fundamental entre uso y mención está basada en la *teoría de la jerarquía* de lenguajes que consiste en distinguir entre un *lenguaje objeto* y el *metalenguaje*, con el cual hablamos acerca del lenguaje objeto⁴.

Actualmente, se ha convenido en distinguir los niveles de lenguaje mediante el uso de comillas. Pero aún así, en nuestro ejemplo, ambos lenguajes usan los signos del idioma español, lo que puede llevar a confusión sobre qué parte del enunciado se usa y qué parte se menciona. Pero si recurrimos a un ejemplo en otro idioma, vemos

³ José FERRATER MORA y Hughes LEBLANC. *Lógica Matemática*. Fondo de Cultura Económica. México, 1955.

⁴ O. DUCROT, y T. TODOROV. *Diccionario enciclopédico de las ciencias del lenguaje*. Buenos Aires. Siglo XXI. 1972.

claramente la distinción entre uso: “La circunferencia es una curva cerrada cuyos puntos son equidistantes de otro, el centro, situado en el mismo plano” y mención: “Circunferencia’ is a Spanish word with five syllables.” En este caso, el español es el lenguaje objeto y el inglés, el metalenguaje.

Los lenguajes de mercado funcionan como metalenguajes para describir un texto y permitirle al “lector” manipularlo. Muchas veces, como en el ejemplo en español sobre el español, se crean a partir del lenguaje objeto que manipulan y por eso es importante la convención del marcado.

El mercado de HTML

Teniendo estas distinciones en consideración, podemos pasar ahora a comenzar a describir los lenguajes de marcado que nos interesan hoy, HTML y XML.

HTML o lenguaje de marcado hipertextual es uno de los más difundidos sistemas de marcado informático que define cómo se presentan en los navegadores de Internet el texto, las imágenes y los sonidos de los documentos publicados.

Cumple con los requisitos de un lenguaje de marcado en tanto:

1. Cuenta con una norma que define lo que constituye un marcado válido, es decir los signos del lenguaje: las **etiquetas**, limitadas por los símbolos **<** y **>**, dentro de los cuales se coloca el nombre de la etiqueta **<nombre de etiqueta>**.
2. Cuenta con una norma que define lo que ese marcado significa: cada etiqueta comunica su propio significado de diseño, asociando una regla que define una **característica de diseño (estructura o estilo)** a la porción de texto que marca. Por ejemplo, la etiqueta **<p>** indica que el texto a continuación es un párrafo.

HTML ha sido tan exitoso porque incluye instrucciones (etiquetas) para interconectar documentos (hipervínculos), de allí que se llame hipertexto, e instrucciones que permiten la interacción del lector (formularios).

Así como el resaltado está destinado a los ojos humanos, el marcado hipertextual está destinado a procesadores de HTML, normalmente un navegador de Internet, que leen las indicaciones de las etiquetas y presentan el texto según las reglas asociadas.

En las siguientes figuras aparece el texto en HTML y su interpretación visual en un navegador:

```
<p><br>
```

Figura 6: Descripción de un archivo gráfico en HTML



The Paradox of Porta Ludovica...
the native lives in a "magic
space" where the directions
front, back, left, and right
are not valid and consequently
all orientation is impossible.

Figura 7: Archivo gráfico interpretado por Netscape.

El hecho de que HTML esté destinado a "lectores" no humanos, es decir programas procesadores de HTML, no significa que los humanos no podamos leerlo. Al contrario de los programas que producen o guardan datos usando un lenguaje binario, HTML utiliza un lenguaje textual. La ventaja es que nos permite leerlo sin necesidad de recurrir a programas especializados. Con un simple procesador de texto como NotePad podemos abrir un archivo HTML. En el ámbito del desarrollo de programas, esto es ventajoso porque los programadores pueden corregir errores mucho más fácilmente que en una codificación binaria.

El procesador de texto nos permite no sólo abrir un archivo HTML sino también *escribirlo*. Sin embargo, al leer o escribir un archivo HTML en un procesador de texto no vemos lo mismo que ve el navegador. Nosotros vemos el archivo HTML y el navegador ve las instrucciones para presentar una página Web, ésta es una diferencia muy importante de recordar. A veces es difícil de entender que lo que escribimos en el procesador de texto o incluso en el editor HTML es a menudo muy diferente de lo que presenta en pantalla el navegador⁵. Veamos otro ejemplo con páginas del sitio The Modern Word.

⁵ Jennifer KYRNIN. "HTML vs. What's On the Screen". En: What You Need to Know About. 13 de enero de 2003.

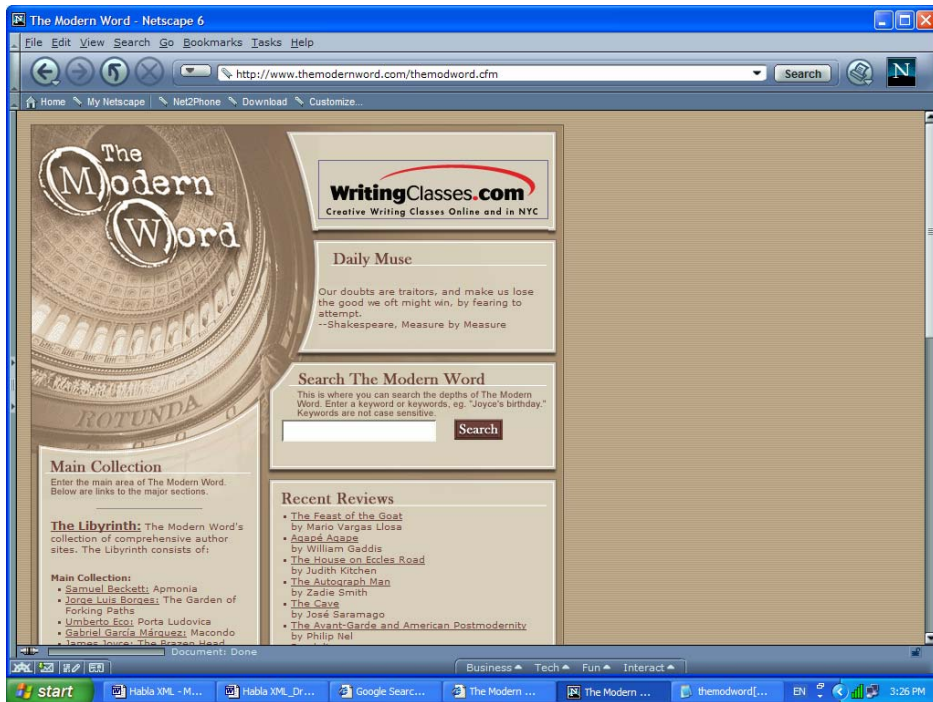


Figura 8: Página de inicio de The Modern Word

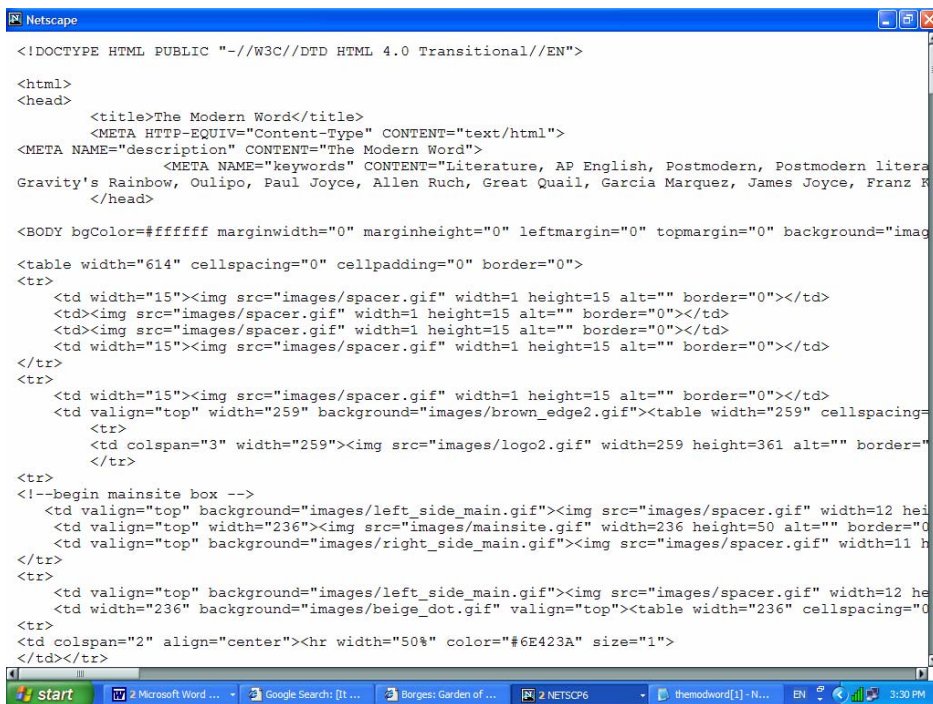


Figura 9: Documento HTML de la misma página de inicio.

Existen muchos editores de HTML como Composer de Netscape o FrontPage de Microsoft, que permiten ver y escribir el archivo HTML casi como lo presenta el

navegador. Se los llaman WYSIWYG, what you see is what you get, lo que ves es lo que obtienes. Lo que hacen es una interpretación simultánea entre el lenguaje de diseño que nosotros comprendemos y el lenguaje de mercado HTML que es el que el navegador comprende. Algunos editores más sofisticados, como Dreamweaver o HomeSite de Macromedia, permiten pasar de una presentación WYSIWYG a una presentación de marcado, lo que le otorga mayor control al creador de la página.

Analicemos algunas de las etiquetas de HTML:

**** especifica la regla que le instruye al navegador presentar en negrita el texto rodeado o “marcado” por esta etiqueta.

<center> le instruye al navegador centrar el texto marcado al presentarlo.

<i> le instruye al navegador presentar en cursiva el texto marcado.

A continuación, presentamos una lista de algunas de las etiquetas más comunes:

ETIQUETA HTML	INSTRUCCIÓN
<HTML></HTML>	Crea un documento HTML.
<head></head>	Demarca información importante sobre la página que no se presenta en el navegador.
<body></body>	Demarca la información visible en el navegador.
<title></title>	Coloca el nombre del documento en la barra de títulos.
<h1></h1>	Crea el título más grande.
<h6></h6>	Crea el título más pequeño.
	Crea texto en negrita.
<i></i>	Crea texto en cursiva.
	Enfatiza una palabra (negrita o cursiva).
	Crea un hipervínculo.
	Crea un vínculo de correo electrónico.
<p></p>	Crea un nuevo párrafo.
	Crea una lista numerada.
	Presenta cada elemento de la lista (les agrega números a las listas ol y viñetas a las ul).
	Crea una lista no numerada con viñetas.
<table></table>	Crea una tabla.
<tr></tr>	Demarca una fila en una tabla.
<td></td>	Demarca una celda en una fila.

Tabla 3: Etiquetas de marcado HTML y su significado.

En el lenguaje escrito distinguimos las palabras separándolas mediante espacios. En la nueva versión de HTML distinguimos las secuencias de marcado abriendo y cerrando cada etiqueta. La etiqueta de apertura se escribe **<nombre de la etiqueta>** y la de cierre se escribe **</nombre de la etiqueta>**.

Existen algunas excepciones en las que basta simplemente con una etiqueta de apertura para indicar una característica de la página Web, como en el caso de la etiqueta **<hr>**, que le indica al navegador crear una línea horizontal a lo ancho de la página. Habiendo una sola palabra no hay necesidad de espacios.



Figura 10: Presentación de la barra horizontal indicada por **<hr>**

Como puede verse en algunos de los ejemplos anteriores, las etiquetas pueden contener *atributos*. Los atributos se definen mediante una *propiedad* a la que se le asigna un *valor*. Por ejemplo, se puede definir el atributo de ancho para la etiqueta **<tabla>** mediante la propiedad **width** y un valor que puede darse en términos de **porcentaje** o **píxeles** (el porcentaje se mide respecto al tamaño total de la pantalla; un píxel es la mínima unidad visual disponible en un medio digital y es por lo tanto una unidad relativa que depende de la resolución de la pantalla). La sintaxis es la siguiente:

<nombre de la etiqueta ATRIBUTO=valor>

Por ejemplo: **<table WIDTH=100%>**

Ya podemos empezar entonces a escribir HTML. Por ejemplo, al escribir en Notepad:

```
<b>Inscripción</b>
```

Internet Explorer, por ejemplo, nos presenta:

```
Inscripción
```

¿Qué pasa si queremos que el texto en negrita aparezca además en cursiva? Debemos considerar una tercera condición fundamental para un lenguaje de marcado (además de definir los signos y sus significados):

3. Se debe contar con una norma que defina cómo pueden combinarse los signos de marcado, es decir una gramática.

Las etiquetas, como las palabras de un idioma, pueden efectivamente combinarse, para obtener instrucciones complejas. Por ejemplo, la secuencia de marcado:

```
<center><b>IV Congreso Latinoamericano de Traducción e Interpretación</b></center>
```

se presenta en un navegador de la siguiente manera:

```
IV Congreso Latinoamericano de Traducción e Interpretación
```

La secuencia siguiente se puede usar para procesar las entradas de una agenda:

```
<ul>
<li>Romina L Marazzato
<ul>
<li>Empresa: Language Compass
<li>Correo electrónico: romina@languagecompass.com
<li>Teléfono: (858) 488-0564
<li>Celular: (858) 331-0564
<li>Fax: (858) 488-0594
<li>Dirección: 1380 Garnet Ave.
```

```

<li>Ciudad: San Diego
<li>Estado: California
<li>Código postal: 92109
</ul>
</ul>
<ul>
<li>Sietske Brown
</ul>
<li>Empresa:
<li>Correo electrónico: sietske@hotmail.com
<li>Teléfono: (831) 642-9494
<li>Celular:
<li>Fax: 831 642-9595
<li>Dirección: 455 Ramona Ave., Apt. 18
<li>Ciudad: Monterrey
<li>Estado: California
<li>Código postal: 93940
</ul>
</ul>

```

Figura 11: Secuencia para entradas de agenda en HTML.

En el procesador de texto esta secuencia también puede escribirse:

```

<ul> <li>Romina L Marazzato </ul> <li>Empresa: Language Compass <li>Correo
electrónico: romina@languagecompass.com <li>Teléfono: (858) 488-0564 <li>Fax: (858)
488-0594 <li>Dirección: 1380 Garnet Ave. <li>Ciudad: San Diego <li>Estado: California
<li>Código postal: 92109 </ul> </ul> <ul> <li>Sietske Brown </ul> <li>Empresa:
<li>Correo electrónico: sietske@hotmail.com <li>Teléfono: (831) 642-9494 <li>Fax: 831
642-9595 <li>Dirección: 455 Ramona Ave., Apt. 18 <li>Ciudad: Monterrey <li>Estado:
California <li>Código postal: 93940 </ul> </ul>

```

Figura 12: Otra manera de escribir la misma secuencia en HTML.

Esta secuencia, al ser leída e interpretada por el navegador para ser presentada en pantalla, asume la siguiente forma:

- Romina L Marazzato
 - Empresa: Language Compass
 - Correo electrónico: romina@languagecompass.com
 - Teléfono: (858) 488-0564
 - Celular: (858) 331-0564
 - Fax: (858) 488-0594
 - Dirección: 1380 Garnet Ave.
 - Ciudad: San Diego
 - Estado: California
 - Código postal: 92109

- Sietske Brown
 - Empresa:
 - Correo electrónico: sietske@hotmail.com
 - Teléfono: (831) 642-9494
 - Celular:
 - Fax: 831 642-9595
 - Dirección: 455 Ramona Ave., Apt. 18
 - Ciudad: Monterrey
 - Estado: California
 - Código postal: 93940

Figura 13: Secuencia para entradas de agenda interpretada por un navegador.

Como puede verse en el ejemplo, la manera de escribir en HTML (de corrido o no) no afecta el resultado en pantalla, pero la primera secuencia es mucho más elegante y permite encontrar errores mucho más fácilmente.

Como mencionamos antes, HTML es un lenguaje “controlado” derivado de SGML, y como el inglés controlado, tiene sus limitaciones. El HTML poseía una gramática bastante laxa, que permitía, por ejemplo, escribir el nombre de las etiquetas en mayúscula o minúscula (como `` o `` para negritas) y combinar el orden de las etiquetas de diversas maneras. Por ejemplo, `<center>Inscripción </center>`, producía el mismo resultado que `<center>Inscripción</center>`.

Además, a medida que las necesidades de la red crecieron, las etiquetas de HTML se hicieron más y más complicadas, combinando etiquetas de estilo y etiquetas de contenido en el mismo lenguaje. Ante esta complicación, el W3C decidió separar las *instrucciones de estilo* de una página Web de las *instrucciones de contenido*. Las etiquetas que definen contenido, como `<h1>`, permanecen en el lenguaje HTML con el que se escribe el documento. En cambio, se prefiere que las etiquetas que definen estilo, como `` (para la tipografía), se utilicen en planillas de estilo asociadas al documento, llamadas hojas de estilo en cascada (CSS)⁶.

Las hojas de estilo⁷ describen las características de estilo de un documento HTML, es decir, indican cómo se presentan las etiquetas de contenido en la pantalla. Definen el diseño de la página, el tipo y estilo de tipografía, y el tamaño de los elementos de texto, tales como encabezados y texto de cuerpo. Para las páginas Web, una hoja de estilo también puede describir el color o imagen de fondo por omisión, los colores de los enlaces de hipertexto y, posiblemente, del contenido de una página. Las hojas de estilo ayudan a asegurar la consistencia en todas las páginas (o un grupo de ellas) en un documento o un sitio Web.

⁶ Hakon WIUM LIE y Bert BOS. *Cascading Style Sheets: Designing for the Web*. 1999. Segunda edición.

⁷ Web Style Sheets Home Page. W3C. <http://www.w3.org/Style/>

La separación de los elementos de diseño visuales de la estructura lógica de una página Web permite controlar el estilo sin comprometer la integridad estructural de los datos.

Esto significa que un mismo documento HTML puede estar asociado a dos hojas de estilo diferentes, una que le indique mostrar el título más grande h1 en un tamaño de tipografía 18 y otra que le indique mostrarlo en tamaño 16 y en mayúsculas.

El concepto de *cascada* tiene que ver con esta posibilidad de múltiples hojas de estilo. Significa que pueden especificarse diferentes hojas de estilo con niveles predeterminados de precedencia para una misma página o grupo de páginas relacionadas. Por ejemplo, en un sitio Web podemos querer que el título h1 se interprete en tipografía 36, mayúsculas y color rojo en la página de inicio, pero no en las demás páginas del sitio.

Además, las CSS cancelan los parámetros predeterminados del navegador. El navegador está programado para hacer una interpretación de las etiquetas de contenido, en caso de que el autor del documento no haya usado una CSS. Pero si la ha usado, entonces se le da prioridad a las características de estilo establecidas por el autor. Las CSS pueden estar publicadas en el documento o no, haciéndose referencia a ellas mediante una especie de hipervínculo.

ETIQUETAS DE ESTILO EN CSS	INSTRUCCIÓN
<code>{font-family: Helvetica, Geneva;</code>	se aplica la primera tipografía disponible de la lista
<code>font-style: italic;</code>	presenta el texto en cursiva
<code>font-weight: bold;</code>	presenta el texto en negrita
<code>font-size: 10pt;</code>	presenta el texto en tipografía tamaño 10.
<code>color: #00FFFF;</code>	presenta el texto en el color hexagesimal #00FFFF, que es una gama de celeste verde agua

Tabla 4: Ejemplos de etiquetas de estilo para el elemento estructural p.

Las CSS también permiten crear grupos de estilo⁸, agregando definiciones de estilo de una clase particular a las definiciones de varias etiquetas diferentes. Las clases se escriben: `.clase {propiedad 1: valor 1; propiedad 2: valor 2}`, con un punto que precede al nombre de la clase.

En la figura siguiente se muestra un ejemplo de definición de clases.

```
<style>
```

⁸ Jakob NIELSEN. "Effective Use of Style Sheets". En: *Alertbox*, julio 1997.

```

p {font-family: sans-serif; font-size: 10pt}
h1 {font-family: serif; font-size: 30pt}
h2 {font-family: serif; font-size: 24pt}
.red {color: red}
.green {color: magenta}
.blue {color: blue}
</style>

```

Figura 14: Definiciones de clases en CSS.

El uso de las etiquetas y las clases puede combinarse, como se muestra en la siguiente figura.

```

<h1 class="red">El texto se presenta en tipografía serif de 30 puntos y color rojo.</h1>

<p class="green">El texto se presenta en tipografía sans-serif de 10 puntos y color magenta.</p>

<p>El texto se presenta en tipografía sans-serif de 10 puntos y el color predeterminado del navegador.</p>

```

Figura 15: Etiquetas y clases combinadas.

La evolución del HTML dio como resultado un lenguaje más estandarizado, aunque sin mayores diferencias, llamado XHTML. Las diferencias entre HTML y XHTML son las siguientes:

XHTML	HTML
Se escribe con minúsculas .	Puede escribirse con MAYÚSCULAS, minúsculas, o uNa Combinación de AmbaS.
Las etiquetas deben abrirse y cerrarse . Las etiquetas únicas, como <hr> e deben contener una barra oblicua de cierre dentro de la misma etiqueta <hr /> .	Las etiquetas no necesitan cerrarse . Si algunas etiquetas de estilo permanecen abiertas, el navegador simplemente interpreta que el estilo debe continuarse hasta encontrarse con otra etiqueta que lo "sobrescriba".
Los atributos deben contener un nombre y un valor . Los valores deben citarse entre comillas . Aquellos que en HTML pueden aparecer sueltos, deben declararse . Por ejemplo: <hr noshade="noshade"> o <table WIDHT=100%> .	Los atributos tienen una sintaxis más laxa y pueden aparecer sueltos. Por ejemplo: <hr noshade>
Las etiquetas se deben anidar en orden. Por ejemplo, la secuencia <i> debe	Las etiquetas no necesariamente deben anidarse en orden. Por ejemplo, la

cerrarse <code></i></code> .	secuencia <code><i></code> puede cerrarse escribiendo <code></i></code> o <code></i></code> .
--	--

Tabla 5: Diferencias entre HTML y XHTML.

Por último, debemos tener en cuenta la sintaxis válida para un documento HTML completo. La estructura básica es la siguiente:

```

<HTML>
<head>
<title>
aquí va el título de la página
</title>
</head>
<body>
aquí va el contenido de la página
</body>
</HTML>

```

Figura 16: Esqueleto de un documento HTML.

Pero dado que hay diferentes maneras de escribir los documentos HTML, debe declararse la especificación del tipo de documento para que el navegador sepa cómo interpretarlo. Una declaración común es:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<HTML>
<head>
<title>
aquí va el título de la página
</title>
</head>
<body>
aquí va el contenido de la página
</body>
</HTML>

```

Figura 17: Ejemplo de declaración del tipo de HTML utilizado para crear un documento.

Resumen de HTML

Recapitulando, sabemos hasta ahora que HTML es un lenguaje de marcado que le da instrucciones al navegador acerca de cómo presentar el texto de un documento HTML en una página Web. Realiza esta tarea utilizando dos tipos de etiquetas: etiquetas de contenido y etiquetas de estilo.

Las etiquetas de contenido definen categorías generales de texto, como encabezado, título, párrafo, tabla, hipervínculo, lista numerada o lista no numerada, etc. Las etiquetas de estilo definen la apariencia de las categorías anteriores, definiendo estilos como negrita, color de la tipografía, justificación,

HTML	Lenguaje de marcado que define cómo se presentan en los navegadores de Internet el texto, las imágenes y los sonidos de los documentos publicados
Condiciones	<ol style="list-style-type: none"> 1) Un marcado válido se define mediante etiquetas limitadas por los símbolos < y >. Cada etiqueta debe abrirse y cerrarse para indicar una secuencia de marcado válida: <nombre de la etiqueta> texto </nombre de la etiqueta>. 2) Cada etiqueta asocia una regla que define una característica de estructura o estilo a la porción de texto que marca. Por ejemplo, <p>: párrafo, <n>: negrita 3) Cuenta con una gramática que define cómo pueden combinarse las etiquetas para producir secuencias de marcado bien formadas. Las secuencias de etiquetas deben abrirse y cerrarse en orden: <center>texto </center> es una secuencia bien formada <center>texto</center> no es una secuencia bien formada
CSS	Mecanismo por el cual se separan las instrucciones de estilo de una página Web de las instrucciones de contenido
Validación	<p>Un documento bien formado debe contener la declaración del tipo de HTML utilizado y la secuencia fundamental de etiquetas de contenido:</p> <pre><HTML> <head> <title> <body></pre> <p>Todos estos elementos están predefinidos en las normas del W3C.</p>

Tabla 6: Características fundamentales de HTML

XML: ¿evolución o revolución?

Al comienzo, dijimos que íbamos a hablar de dos dialectos de SGML, HTML y XML. Comparamos el HTML al inglés controlado y en este sentido, pueden haber escuchado que el XML es como el HTML, pero con más libertad, ya que se pueden

crear secuencias y etiquetas menos rígidas, así como el inglés permite crear expresiones idiomáticas vedadas en inglés controlado.

Como toda analogía, esta comparación tiene sus limitaciones y no es precisa. En todo caso, si comparamos el HTML al inglés controlado, podemos comprar al XML con un hipotético idioma germánico, es decir, un idioma, derivado de un mismo tronco común, el indoeuropeo, pero más productivo (en el sentido de la generación de lenguas) y más abarcativo. El SGML sería comparable al indoeuropeo, ya que es la lengua madre.

En términos generales, XML es por un lado, el lenguaje en el que se basa la nueva versión de HTML. Se deriva también del SGML, pero es menos estricto y brinda la posibilidad de crear lenguajes específicos para trabajar en diferentes disciplinas. Es decir, es un lenguaje de marcado para crear lenguajes de marcado.

XML significa lenguaje de marcado extensible. Esto quiere decir que no cuenta con una lista predeterminada de etiquetas. Las etiquetas pueden crearse según las necesidades del autor. Esta flexibilidad de XML permite crear lenguajes aplicables a campos muy diversos.

XML ha permitido generar lenguajes de marcado estandarizados y difundidos, como MathML, para trabajar en matemáticas, SMIL, para trabajar en multimedios (sus siglas significan "lenguaje de integración para multimedios sincronizados", de Synchronized Multimedia Integration), y CML o lenguaje de marcado químico, para manipular información molecular.

Sin embargo el poder de XML surge de la posibilidad de crear lenguajes originales. Por ejemplo, podemos crear categorías, llamadas formalmente *elementos*, como <padre>, <abuelo>, <hermana>, etc., para trabajar en genealogía o <receta>, <autor>, <ingredientes>, para cuestiones culinarias.

La sintaxis general de XML es similar a la de XHTML:

Sintaxis para XML	Ejemplos
Las etiquetas deben abrirse y cerrarse.	<code><p>This is another paragraph</p></code>
Las etiquetas deben anidarse correctamente.	<code><i>This text is bold and italic</i></code>
Los nombres de etiquetas son sensibles a la diferencia entre mayúsculas y minúsculas.	<code><Message>This is incorrect</message></code> <code><message>This is correct</message></code>
Los atributos deben citarse entre comillas.	<code><memo fecha="17/11/2002"></code>
Todo documento de XML debe contener un elemento raíz dentro del cual se encuentren los subelementos, que deben	<code><raíz></code> <code><subelemento padre></code> <code><subelemento hijo></code>

anidarse correctamente.	<pre> </subelemento hijo> </subelemento padre> </raíz> </pre>
-------------------------	---

Tabla 7: Elementos fundamentales de la sintaxis de XML

Como en XHTML, los documentos XML deben contener la *declaración de XML*, que define la versión de XML y la codificación de caracteres empleadas (esto determina el idioma y el alfabeto a usar, así como la codificación binaria del mismo).

```

<?XML version="1.0" encoding="ISO-8859-1"?>
<memo fecha="17/11/2002">
<a>Jimena</a>
<de>Alvarez</de>
</memo>

```

Figura 18: Ejemplo de declaración del tipo de XML que utiliza el documento

En este ejemplo, el documento cumple con la especificación 1.0 de XML del W3C y utiliza el set de caracteres ISO-8859-1, es decir Latin-1/europa occidental.

Veamos cómo procesa XML nuestro ejemplo de una entrada de agenda:

```

<?XML version="1.0" encoding="ISO-8859-1"?>
<persona>
<nombre>Romina L Marazzato</nombre>
<empresa>Language Compass</empresa>
<email>romina@languagecompass.com</email>
<tel tipo="laboral">(858) 488-0564</tel>
<tel tipo="celular">(858) 331-0564</tel>
<tel tipo="fax">(858) 488-0594</tel>
<direccion>1380 Garnet Ave.</direccion>
<ciudad>San Diego</ciudad>
<estado>California</estado>
<codigopostal>92109</codigopostal>
</persona>

```

Figura 19: Ejemplo de entrada de agenda escrita en XML.

Al abrir este documento en un navegador de Internet, el navegador aplicará la hoja de estilo por defecto para XML y aparecerá en pantalla lo siguiente:

```

<persona>
<nombre>Romina L Marazzato</nombre>
<empresa>Language Compass</empresa>
<email>romina@languagecompass.com</email>

```

```

<tel tipo="laboral">(858) 488-0564</tel>
<tel tipo="celular">(858) 331-0564</tel>
<tel tipo="fax">(858) 488-0594</tel>
<direccion>1380 Garnet Ave.</direccion>
<ciudad>San Diego</ciudad>
<estado>California</estado>
<codigopostal>92109</codigopostal>
</persona>

```

Figura 20: Presentación de la entrada de agenda XML en un navegador

Es decir, no obtenemos el resultado visual de HTML. Pero podemos ver el documento bien formado para decidir qué tipo de resultado visual queremos obtener.

Como en cualquier lengua natural, cuando un documento XML utiliza la sintaxis correctamente, se dice que está bien formado. Por otro lado, el documento puede legitimarse con una definición de tipo de documento o DTD. Cuando un documento XML se valida con una DTD, se dice que es un documento válido.

Para que un documento se valide frente a una dada DTD, debemos agregar al mismo una etiqueta que indique la DTD a utilizar:

```

<?XML version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE entrada SYSTEM "Entradas.dtd">
<persona>
  <nombre>Romina L Marazzato</nombre>
  <empresa>Language Compass</empresa>
  <email>romina@languagecompass.com</email>
  <tel tipo="laboral">(858) 488-0564</tel>
  <tel tipo="celular">(858) 331-0564</tel>
  <tel tipo="fax">(858) 488-0594</tel>
  <direccion>1380 Garnet Ave.</direccion>
  <ciudad>San Diego</ciudad>
  <estado>California</estado>
  <codigopostal>92109</codigopostal>
</persona>

```

Figura 21: Declaración de una DTD para validar un documento XML.

La DTD define los elementos legales de un documento. Es decir, define la estructura del documento mediante una lista de los elementos que pueden usarse en el mismo y facilita el intercambio de datos.

Los componentes de una DTD son:

Componente de DTD	Definición	Ejemplo
-------------------	------------	---------

Elementos	Componentes principales.	<code><persona></code> (en XML) <code><body></code> (en HTML)
Etiquetas	Marcan los elementos. Delimitan cada elemento mediante una etiqueta de <apertura> y una de </cierre>.	<code><nombre>n</nombre></code>
Atributos	Brindan información adicional sobre los elementos. Deben consistir en una <i>propiedad</i> y un <i>valor</i> , con la siguiente sintaxis <elemento propiedad="valor">	<code><tel tipo="celular">t</tel></code>
Entidades	Variables que definen texto del lenguaje objeto para que no se confunda con el metalenguaje.	<code>&lt;</code> ; elicit el carácter < <code>&gt;</code> ; elicit el carácter > <code>&amp;</code> ; elicit el carácter & <code>&quot;</code> ; elicit el carácter " <code>&apos;</code> ; elicit el carácter '
CDATA (Character Data)	Información de caracteres que no es texto de marcado y no es analizada por un analizador de sintaxis. La información de caracteres consiste en texto que no es texto de marcado en sí mismo. Generalmente, se encuentra entre la etiqueta de apertura y la de cierre de un elemento. Las etiquetas dentro de CDATA no se tratan como marcado y las entidades no se expanden.	<code><![CDATA[</code> <code><saludo></code> <code>¡Hola!</code> <code></saludo></code> <code>]]></code> <code><saludo></code> y <code></saludo></code> se tratan como texto CDATA y no como etiquetas de marcado
PCDATA (Parsed Character Data)	Información de caracteres analizada por el analizador de sintaxis. Las etiquetas dentro de PCDATA se tratan como marcado y las entidades se expanden.	

Tabla 8: Componentes de la DTD para XML.

En nuestro ejemplo de entrada de agenda, una posible DTD sería:

```

<?XML version="1.0"?>
<!DOCTYPE entrada [
  <!ELEMENT persona
(nombre,empresa,email,tel,direccion,ciudad,estado,codigopostal,foto)
  <!ELEMENT nombre (#PCDATA)
  <!ELEMENT empresa (#PCDATA)
  <!ELEMENT email (#PCDATA)
  <!ELEMENT tel (#PCDATA)
  <!ELEMENT dirección (#PCDATA)

```



```

<!ELEMENT ciudad (#PCDATA)
<!ELEMENT estado (#PCDATA)
<!ELEMENT codigopostal (#PCDATA)
<!ATTLIST tel tipo (laboral|cellular|fax) CDATA #IMPLIED
]>

```

Figura 22: Ejemplo de una DTD para una entrada de agenda.

Debido a que las DTD pueden definirse de diversas maneras, el W3C ha propuesto una alternativa para definir los elementos legales de un documento llamada esquema de o XML Schema. La definición de elementos válidos de un documento XML mediante el uso de esquemas se conoce como XSD, definición de esquemas de XML. La sintaxis de un esquema XML difiere de la de una DTD pero la función del esquema es la misma que la de una DTD.

Un esquema XML define los elementos y atributos legales de un documento, su cantidad, parentesco, orden y tipo de contenido. Permiten el uso de data types y namespaces. Los primeros permiten escribir el tipo de contenido, como cadenas, número entero, número decimal, etc. Los segundos permiten evitar conflictos entre diferentes documentos que pueden usar el mismo nombre de elemento con funciones diferentes. Los nombres de elementos se asocian a una definición única mediante un enlace URI (identificador uniforme de recursos), que es como un hipervínculo.

La principal ventaja de los esquemas consiste en que se escriben en XML, por lo que no es necesario utilizar reglas diferentes.

En nuestro ejemplo de entrada de agenda, un esquema de XML guardado en un archivo entrada.xsd se vería de la siguiente manera:

```

<?XML version="1.0" encoding="ISO-8859-1" ?>
<xs:schema
XMLns:xs="http://www.w3.org/2001/XMLSchema">

<!-- definition of simple elements -->
<xs:element name="nombre" type="xs:string"/>
<xs:element name="empresa" type="xs:string"/>
<xs:element name="email" type="xs:string"/>
<xs:element name="tel" type="xs:string"/>
<xs:element name="direccion" type="xs:string"/>
<xs:element name="ciudad" type="xs:string"/>
<xs:element name="estado" type="xs:string"/>
<xs:element name="codigopostal" type="xs:string"/>

<!-- definition of attributes -->
<xs:attribute name="tipo" use="optional">
  <xs:simpleType>
    <xs:restriction base = "xs:string">

```

```

        <xs:enumeration value="laboral"/>
        <xs:enumeration value="celular"/>
        <xs:enumeration value="fax"/>
    </xs:restriction>
</xs:simpleType>
</attribute>

<!-- definition of complex elements -->
<xs:element name="persona">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="nombre"/>
            <xs:element ref="empresa"/>
            <xs:element ref="email"/>
            <xs:element ref="tel"/>
            <xs:element ref="dirección"/>
            <xs:element ref="ciudad"/>
            <xs:element ref="estado"/>
            <xs:element ref="codigopostal"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>

```

Figura 23: Ejemplo de una XSD para una entrada de agenda.

La diferencia entre HTML es enorme, a pesar de que ciertas convenciones de lenguaje se mantengan. Esto ha hecho que muchos seguidores de XML no lo consideren una simple evolución de HTML sino una revolución. Entre los conocedores del tema se bromea diciendo que HTML es WYSIWAG, lo que ves es TODO lo que obtienes, mientras que XML, aparentemente más oscuro, tiene un poder mucho mayor⁹.

Por último, cabe mencionar que dicho poder está asociado a un conjunto de normas muy estrictas, por lo que cualquier error en un documento XML llevará a una falla del programa que lo esté procesando. Este es un concepto muy importante a tener en cuenta a la hora de traducir. Al trabajar con archivos XML se debe tener sumo cuidado de no alterar la sintaxis del metalenguaje para no corromper el documento. Por otra parte, al trabajar con herramientas de traducción que procesen archivos XML, muchas nos encontramos con fallas que no son culpa del traductor sino del creador del documento.

Presentación de archivos XML

⁹ Charles F. GOLDFARB y Paul PRESCOD. The XMI Handbook. Prentice Hall. Nueva Jersey. 2000.

Anteriormente dijimos que los documentos XML no pueden ser presentados directamente por un navegador de Internet como lo son los HTML, dado que no existe una hoja de estilo predeterminada con elementos hijos.

Para poder presentar visualmente de manera elegante un documento XML es necesario hacer uso de una hoja de estilo. La presentación visual puede tener como objetivo una pantalla, un documento impreso o, incluso, un producto de audio.

Para un mismo documento XML se pueden especificar diversas CSS, según el uso que se le vaya a dar al documento. Las CSS se expresa, como en HTML, mediante una serie de reglas que se guardan en un archivo de texto al que se hace referencia en el archivo XML. Cada regla contiene el nombre del elemento al que se aplica y el estilo definido¹⁰.

Sin embargo, las CSS tienen una eficacia limitada y el W3C recomienda el uso de un lenguaje de estilo más sofisticado y apropiada para XML, llamado XSL o lenguaje de estilo extensible. La característica más importante de XSL es que permite transformar un documento XML en otro documento XML, en un documento HTML u otro tipo de documento para que pueda ser reconocido por un navegador, una hoja de cálculo, un programa de diseño, etc.

Las definiciones de estilo que hacen uso de XSL también se guardan en un archivo de texto al que se hace referencia desde el archivo XML.

```
<?XML version="1.0" encoding="ISO-8859-1"?>
<?XML-stylesheet type="text/xsl" href="menu.xsl"?>
<menu>
  <item>
    <nombre>Flan mixto</nombre>
    <precio>$1,80</precio>
    <descripcion>
      Flan de huevo acompañado de crema chantillí y dulce de leche
    </descripcion>
  </item>
</menu>
```

Figura 24: Referencia a un archivo de estilo XSL.

Resumen de XML

¹⁰ F. Javier GARCÍA CASTELLANO. Cascading Style Sheets. Tutorial de Hojas de Estilo en Cascada (CSS) para XML. Versión 2.0. Marzo 2003.

XML	<p>Lenguaje de marcado extensible que permite crear lenguajes específicos a diversos campos, ya que describe el contenido de los datos que marca.</p> <p>No está predefinido, por lo que se deben crear todas las etiquetas a usar. Sin embargo, existen aplicaciones estandarizadas como CML.</p>
Condiciones	<p>Un marcado válido se define mediante etiquetas que contienen elementos, atributos, declaraciones, comentarios y enlaces.</p> <p>Las etiquetas no definen la presentación o estilo del texto marcado sino que describen su contenido.</p> <p>Las etiquetas deben abrirse y cerrarse, anidarse correctamente y escribirse en minúsculas. Todo documento de XML debe contener un elemento raíz dentro del cual se encuentren los subelementos. Los elementos pueden contener atributos.</p>
XSL	<p>Lenguaje de estilo extensible que permite crear interpretaciones de estilo a fin de transformar un documento XML en otro documento que pueda ser interpretado y presentado mediante un navegador, una hoja de cálculo, un programa de diseño, etc.</p>
Validación	<p>El documento puede validarse frente a una DTD que define los componentes legales de un documento (elementos, etiquetas, atributos, entidades, PCDATA, CDATA).</p> <p>El W3C recomienda el uso de esquemas XML o XSD (XML schema definition). El esquema define los componentes legales de un documento (elementos y atributos legales de un documento, su cantidad, parentesco, orden y tipo de contenido) usando las mismas convenciones que XML.</p> <p>Ambos mecanismos son autodescriptivos.</p>

Tabla 9: Características fundamentales de XML.

Diferencias entre XML y HTML

Las diferencias fundamentales entre XML y HTML residen en la aplicación de cada uno. Mientras que HTML se emplea para dar una interpretación visual de los datos que marca, XML se emplea para dar una interpretación semántica.

Estos dos lenguajes representan los dos tipos básicos de lenguajes de documentos digitales: procedural y estructural. El marcado procedural describe la forma y el significado de las operaciones tipográficas a las que serán sometidos cada uno de los elementos del documento para generar una presentación visual. El marcado

estructural describe únicamente la estructura lógica del documento digital, sin ocuparse en absoluto de la presentación.

Además, las operaciones tipográficas en HTML son fijas, están predeterminadas por la definición de tipo de documento HTML 4.0 publicada por el W3C. Las estructuras lógicas empleadas en XML son infinitas, no hay limitaciones para las categorías de elementos que pueden crearse, aunque sí existen estructuras predefinidas para aplicaciones estandarizadas.

Veamos a continuación un resumen de las diferencias más importantes.

XML	HTML
Es extensible.	Está predefinido.
Se define mediante la creación de una DTD o una XSD que puede contener elementos totalmente originales.	Se define mediante una DTD cuya lista de componentes válidos es fija.
No reemplaza a HTML, lo complementa.	
Fue concebido para describir datos.	Fue concebido para presentar datos.
Importa el tipo de información.	Importa cómo se ve la información.
Es un lenguaje estructural.	Es un lenguaje procedural.
Tiene una sintaxis estricta.	Tiene una sintaxis estricta, ahora basada en XML, cuyas convenciones son similares.
Es una herramienta autónoma, es decir independiente del hardware, software y plataforma que se utilice. Puede crearse software para su procesamiento fácilmente.	HTML es relativamente independiente pero está concebido para ser interpretado por un navegador.

Tabla 10: Diferencias fundamentales entre HTML y XML.

Conclusión

El objetivo de esta presentación ha sido familiarizar al traductor técnico con los lenguajes de marcado más relevantes en el mundo de la traducción.

Presentamos el concepto de lenguaje de marcado como metalenguaje que permite dar instrucciones acerca del lenguaje objeto del que habla. Definimos al HTML como un lenguaje de marcado procedural que se concentra en el resultado visual y XML como un lenguaje de marcado estructural que se concentra en la estructura lógica de un documento.

Comprender el funcionamiento de ambos lenguajes le permitirá al traductor trabajar con archivos HTML y XML sin confusión. Los elementos constitutivos fundamentales de los signos válidos de cada lenguaje son similares (<, >, nombre de etiqueta, etc.) Esto le permitirá al traductor relevar un documento y poder determinar

qué secciones son parte del metalenguaje y qué secciones son parte del lenguaje objeto y debe, por lo tanto, traducirlas.

Asimismo, un conocimiento conceptual, aunque básico de los lenguajes de marcado y especialmente de XML le permitirá al traductor hacer frente a problemas relacionados con el procesamiento de XML en herramientas de traducción, dándole la posibilidad de encontrar errores y de hacer preguntas informadas.

Otro aspecto importante, que nos quedará para otra oportunidad, es entender el mecanismo de funcionamiento de muchas herramientas de traducción u otro tipo de aplicaciones informáticas que hacen uso del lenguaje XML para el procesamiento interno de los datos.

Nota final: a lo largo del artículo se cita a pie de página bibliografía útil referida a los temas tratados antes de cada llamada.