

---

# Los sitios y el contenido web

---

PID\_00250669

Manuel Mata Pastor



# Índice

<b>Introducción</b> .....	5
<b>Objetivos</b> .....	6
<b>1. Sitios web estáticos frente a sitios web dinámicos: repercusiones técnicas y profesionales</b> .....	7
<b>2. Las estrategias habituales</b> .....	9
<b>3. Las herramientas esenciales</b> .....	11
3.1. Editores de texto .....	11
3.2. Aplicaciones ofimáticas y de autoedición .....	12
3.3. Aplicaciones de diseño web .....	13
3.4. Herramientas de traducción asistida .....	14
3.5. Herramientas de traducción de páginas web .....	15
3.6. Otras herramientas .....	16
<b>4. El proceso y sus fases principales</b> .....	18
4.1. Los elementos textuales .....	22
4.2. Los elementos metatextuales .....	23
4.3. Los elementos interactivos .....	25
4.4. Los elementos gráficos .....	28
4.5. Otros elementos localizables .....	31
<b>Resumen</b> .....	33
<b>Bibliografía</b> .....	35



## **Introducción**

En este módulo, y en los ejercicios y actividades que lo acompañan, se estudian los principales aspectos de la localización de sitios y contenido web, servicio en el que empresas y profesionales autónomos encuentran hoy una rentable línea de negocio o una prometedora oportunidad laboral.

Además, se persigue que el estudiante se familiarice con algunas nociones básicas de diseño web. Así, estará en mejor disposición de entender las peculiaridades de los sitios que suelen localizarse y de ofrecer un servicio que, amén de la traducción propiamente dicha, incluya todos los ajustes necesarios para localizar íntegramente un sitio web estático de mediana complejidad.

Internet se ha convertido en el medio de información y comunicación por excelencia de este nuevo siglo y, con ello, en una fuente inagotable de trabajo para los proveedores de servicios lingüísticos. Empresas de toda índole, instituciones y particulares se asoman a la Red de redes para darse a conocer, vender sus productos y servicios e interactuar con sus clientes. Esto se traduce en un ingente volumen de contenidos que, en muchos casos, requiere ser vertido a otros idiomas y adaptado a los mercados y destinatarios a los que van dirigidos.

## Objetivos

Tras la lectura de este módulo y la realización de los ejercicios y actividades que lo acompañan, el estudiante debería haber alcanzado los siguientes objetivos:

1. Conocer algunos conceptos básicos de diseño web, que son necesarios para identificar las peculiaridades y la estructura de un sitio web.
2. Entender las implicaciones técnicas, comerciales, lingüísticas y culturales que llevan aparejadas la localización y la internacionalización de un sitio web.
3. Comprender las principales diferencias entre un sitio web estático y uno dinámico, así como las repercusiones técnicas y profesionales de esta diferenciación.
4. Aplicar las estrategias habituales de localización de un sitio web.
5. Conocer las herramientas que se emplean para localizar un sitio web y sus principales ventajas e inconvenientes.
6. Familiarizarse con los principales elementos que integran un sitio web y con los formatos de archivo y tecnologías más habituales.
7. Localizar los elementos textuales, metatextuales, interactivos y gráficos típicos de un sitio web estático de mediana complejidad.

## 1. Sitios web estáticos frente a sitios web dinámicos: repercusiones técnicas y profesionales

Los abundantes intentos por **taxonomizar** una realidad tan heterogénea como la del contenido web y sus posibles mutaciones se basan en infinidad de criterios. Sin embargo, como instrumento de comunicación y de interacción que es un sitio web, los factores que suelen determinar en mayor medida la forma y el contenido de una web son los objetivos que persigue y los destinatarios a quienes va dirigida, así como la distancia y la relación que median entre estos y sus creadores o titulares.

No obstante, desde el punto de vista de la localización, el aspecto que más importa es, sin duda, el **grado de dinamismo de un sitio web**, pues condiciona a la postre las estrategias, las herramientas y el planteamiento adoptados para localizarlo, así como el coste de hacerlo.

Si atendemos a cómo se ha planteado la interacción entre servidores y clientes y cómo se articula la **proyección** en estos últimos de los contenidos alojados en los primeros, cabe distinguir en líneas generales entre webs estáticas y dinámicas. En las **estáticas**, los contenidos albergados en el servidor de páginas web se transfieren tal cual al cliente que los solicita, de manera que el navegador los interpreta y se los proyecta al usuario. Hoy, sin embargo, es frecuente que haya en el servidor aplicaciones que consultan bases de datos para intercambiar información con el navegador del cliente. En tales casos, la plétora de tecnologías y arquitecturas posibles es inacabable, y esto complica sobremanera el proceso de localización.

Siguen existiendo millares de webs estáticas desde el punto de vista de su arquitectura cliente-servidor, pero que incluyen también elementos dinámicos. En la mayoría de las webs con una arquitectura cliente-servidor estática, suele ser posible **descargar íntegramente sus contenidos** con programas de tele-descarga o navegación fuera de línea (y mantenerlos actualizados en sucesivas sesiones). Esto puede resultar muy útil para disponer de los contenidos localizables de manera casi inmediata, por ejemplo, para analizarlos y presentarle una propuesta de servicios de localización a un posible cliente por iniciativa propia.

En cambio, hoy por hoy, la mayoría de los sitios webs son dinámicos y se diseñan, organizan y mantienen mediante plataformas denominadas **sistemas de gestión de contenidos o CMS (Content Management Systems)**. Estas aplicaciones de servidor procesan contenidos cuya descarga íntegra resulta prácticamente inviable por estar almacenados en bases de datos difícilmente accesi-

### Programas de tele-descarga

Los programas de tele-descarga también se denominan «aspiradores de web». Algunos de los más conocidos son HT-Track, Teleport Pro o WebReaper, entre muchos otros.

bles a distancia. En tales casos, la única manera de descargar tales contenidos para localizarlos —o siquiera para cuantificarlos con el fin de elaborar un presupuesto— es solicitarle al cliente que los extraiga de las bases de datos que los albergan y los facilite en algún formato ofimático convencional (archivo de texto CSV, hoja de cálculo, tabla en documento con formato, etc.) o bien en un formato estándar de intercambio (como XLIFF). La mayoría de los actuales CMS permiten exportar —y reimportar— los contenidos de las bases de datos con la ayuda de complementos o *plug-ins* diseñados con ese propósito (como VPML para WordPress) o mediante el propio gestor de bases de datos (como MySQL).

Así pues, en líneas generales, la caracterización de una web como estática/dinámica, desde el punto de vista de su arquitectura cliente-servidor, constituye el factor al que de manera más drástica queda supeditada la estrategia de localización aplicable. A riesgo de simplificar, desde esta perspectiva, cabría reducir los tipos de webs a tres:

Tipos de web según su arquitectura cliente-servidor y sus componentes

Tipos de web	Web estática	Web semi-estática	Web dinámica
Frecuencia de actualización	Baja	Variable	(Muy) alta
HTML	X	X	X
Contenido gráfico	X	X	X
Contenido multimedia	[X]	X	[X]
Scripts	[X]	X	X
Applets y aplicaciones en cliente	-	X	X
Aplicaciones en servidor	-	-	X
Bases de datos	-	-	X

#### HTML

El HTML no es ni mucho menos el único lenguaje de salida mediante el cual se proyectan los contenidos web en un navegador, aunque sí el más frecuente. La última versión disponible es HTML5. Su implantación dista de ser generalizada, pero ha supuesto, junto con la actualización del lenguaje de hojas de estilo CSS3 y la revitalización del lenguaje JavaScript y sus derivados, un considerable salto cualitativo respecto a tecnologías anteriores.



## 2. Las estrategias habituales

Hasta hace pocos años, la localización de sitios web planteaba, a grandes rasgos, dificultades muy similares a las que en sus inicios tuvo que enfrentar la localización de software. En esencia, se reducían a idear procedimientos para diferenciar de manera (semi)automática **texto localizable** y **código informático**, y separarlos preservando la integridad de este, o bien para extraer el texto localizable de modo que pudiera traducirse y reinsertarse posteriormente en el código. Estas dos siguen siendo las estrategias más comúnmente empleadas: o bien se traducen los textos en su formato original, o **nativo**, o bien se extraen, exportan y/o convierten a un formato intermedio, o de intercambio, que permita localizarlos con alguna herramienta, de modo que luego puedan volver a reinsertarse en su contenedor original.

En principio, la localización de una web puede llevarse a cabo en cualquiera de las etapas de su ciclo vital. Idealmente, aunque no sea lo más habitual, puede plantearse desde su concepción inicial o en las fases de diseño. No obstante, lo más frecuente es que la posibilidad de localizar una web se plantee en algún estadio avanzado de su desarrollo o, sobre todo, una vez finalizada la producción y publicación de su primera versión o de alguna posterior.

En general, cuanto más temprana sea la fase del ciclo vital de una web en la que se acometa su localización, o siquiera se baraje la posibilidad de localizarla, menor será la problemática que esta plantee y mayores serán las oportunidades de internacionalizarla *a priori*.

Disponer de los contenidos íntegros de una web, tanto para su análisis y cuantificación como para su posterior procesamiento y localización, es una tarea que está supeditada, entre otros factores, a los ritmos de actualización de la web en cuestión, a la naturaleza estática o dinámica de esta y a la relación que se mantenga con el cliente. Como en otros ámbitos tradicionales, la traducción suele presupuestarse por palabras contabilizadas en el idioma original y, como en proyectos de localización de software, casi todos los demás conceptos (tareas de ingeniería, diseño gráfico, testeo, etc.) se cuantifican y facturan por horas de trabajo.

### Lecturas recomendadas

A quienes se inician en el terreno de la localización de sitios web, pueden resultarles útiles, entre otros, trabajos como el suplemento monográfico «Global Web» del número 55 de la revista *Multilingual Computing & Technology*, como introducción general, o el artículo de Galibert (2004) sobre la traducción de archivos HTML, además de las páginas del manual de Esselink (2000) dedicadas a los lenguajes de etiquetado.

Uno de los primeros retos técnicos que se planteó en los inicios de internet fue la diferenciación y extracción automatizadas de código informático (etiquetado y elementos interactivos autónomos o imbricados en este) y contenido textual (incluido el metatextual), para poder cuantificar los volúmenes de palabras susceptibles de ser traducidos. Hoy día, prácticamente cualquier herramienta de localización o de traducción asistida (utilidades de recuento de palabras, gestores de memorias de traducción o programas de gestión de proyectos de traducción) resuelve con eficacia este problema: separa del código y contabiliza discriminadamente el contenido textual y metatextual, y detecta posibles repeticiones, coincidencias totales o parciales, etc.

En la actualidad, como se pone de manifiesto cotidianamente en los foros profesionales de traducción y localización, la mayor problemática viene propiciada por el hecho de que la mayoría de las webs son esencialmente estáticas. A efectos prácticos, ya no es tan sencillo obtener los contenidos localizables teledescargándolos, como sí solía ser viable con las webs estáticas —y, por cierto, continúa siéndolo con los elementos o secciones que a menudo siguen conteniendo las actuales webs dinámicas—.

Como es lógico, esta circunstancia merma las posibilidades (técnicas, en primera instancia, y comerciales, por ende) de preparar ofertas de servicios de localización en las que se incluya una estimación económica detallada. Sin embargo, tampoco supone mayor novedad tener que solicitarle al cliente, para su cuantificación, análisis y evaluación los materiales susceptibles de ser localizados. Así ha sucedido siempre en cualquier encargo convencional de traducción y también ha sido —y sigue siendo— el caso en los proyectos de localización de software.

#### FreeBudget y WebBudget

Fueron pioneras en este asunto las herramientas FreeBudget y WebBudget, desarrolladas por la empresa catalana Aquino Software ([www.webbudget.com](http://www.webbudget.com)). Como se deduce de su denominación comercial, la primera nació —y aún subsiste— como utilidad de *freeware* especializada en el recuento pormenorizado y la presupuestación de documentos en formatos convencionales, mientras que la segunda, más reciente y evolución de aquella, surgió como respuesta al reto técnico al que aquí nos referimos.

### 3. Las herramientas esenciales

Atendiendo a las estrategias mencionadas en el apartado anterior, a continuación se analizan, en primer lugar, los principales tipos de herramientas que se emplean para localizar contenido web, haciendo hincapié en sus ventajas e inconvenientes; en segundo lugar, mencionamos algunas otras que permiten extraer los elementos localizables a otros formatos para poder localizarlos y reinsertarlos posteriormente en los archivos de los que proceden. En general, dependiendo de las características del proyecto, los requisitos del cliente, las previsiones de actualización y la consabida naturaleza total o mayoritariamente estática/dinámica de la web, etc., será más recomendable optar por un determinado tipo de herramienta, aunque lo ideal es conocerlas todas y utilizarlas de manera complementaria, atendiendo a las virtudes y carencias de cada una de ellas.

Esselink (2000, págs. 213-219) analiza algunas de las herramientas empleadas en la localización de archivos HTML y XML —y, por extensión, de la inmensa mayoría de los formatos dinámicos que se emplean en la actualidad—, y las agrupa en **cuatro familias**: editores WYSIWYG, editores de texto, gestores de memorias de traducción, y editores que aíslan y protegen el etiquetado. Partiendo de sus observaciones, a continuación reformulamos su análisis y lo ampliamos.

#### 3.1. Editores de texto

Resulta inconcebible traducir una gran cantidad de texto o de archivos etiquetados en una herramienta de este tipo, tanto por los inconvenientes que se mencionan a continuación como por carecer de las prestaciones que hoy ya ofrece cualquier sistema de gestión de memorias de traducción o herramienta de localización para aumentar la productividad, garantizar la coherencia, etc. Sin embargo, el empleo de editores de texto (siempre y cuando cuenten con prestaciones avanzadas) se torna imprescindible como herramienta complementaria para comprobaciones o retoques de texto o código informático, para lo cual habrá que contar con unos conocimientos básicos de la sintaxis del lenguaje en cuestión. Todos los navegadores interactúan con algún visor/editor de texto, propio o externo, que se activa al acceder al código fuente de la página. Las versiones más recientes de los navegadores más conocidos vienen provistas de potentes herramientas diseñadas para desarrolladores pero de extraordinaria utilidad también para que el localizador pueda realizar todo tipo de comprobaciones y retoques, individualmente en archivos sueltos o a gran escala en centenares de ellos, puesto que de —ligeros— archivos de texto plano se trata.

#### Listas de herramientas

Existen en la Red infinidad de listas de herramientas de las categorías a las que aquí nos referimos, comentadas o no, y unas más completas y actualizadas que otras. Junto con repertorios de herramientas clásicos, como el célebre Compendium of Translation Software de la European Association for Machine Translation (cuya última actualización está fechada en marzo del 2010), pueden consultarse otros monográficos o especializados en el sector de la localización, como los de Open-tag, I10nGurus.com, el proyecto ELECT o el Localisation Research Centre, por citar solo algunos de los más conocidos.

#### Contenidos dinámicos

Además de los lenguajes de etiquetado (como HTML o XML), en las webs dinámicas se emplean otros formatos con unas peculiaridades y sintaxis propias. Tal es el caso de los archivos en lenguajes y formatos como ASP, ColdFusion, Go, JSON, JSP o PHP, entre otros. Sin embargo, todos ellos comparten con los formatos tradicionales (HTML, XML, CSS, JS...) su condición de ser archivos de texto sin formato, o sea, plano (*text only*). Por tanto, las herramientas y maneras de procesarlos para localizarlos no difieren, en esencia, de las que se enumeran y analizan en este apartado; ya sea de forma directa o con la ayuda de filtros definidos *ad hoc*.

## Ventajas e inconvenientes de los editores de texto

Ventajas	Inconvenientes
<p>Fácil acceso al código para pequeños ajustes y retoques.</p> <p>No presentan los inconvenientes de las herramientas ofimáticas y de creación y diseño web, en lo que a reinterpretación del código se refiere.</p>	<p>Entorno no gráfico, incómodo y poco atractivo.</p> <p>Riesgo de alteración accidental del código.</p> <p>Riesgo de no detección de texto traducible.</p> <p>Riesgo de traducir código (variables, comentarios, etc.).</p> <p>Carencia de funciones avanzadas de tratamiento de textos (correctores ortográfico y gramatical, búsqueda y sustitución, macros, etc.).</p> <p>Carencia de funciones de gestión de terminología y memorias de traducción y de traducción asistida.</p> <p>Conversión de caracteres especiales, entidades, etc.</p>

A pesar de los inconvenientes anteriores, hoy existen infinidad de **editores de texto avanzados** que, además de permitir procesar los formatos más habituales de contenido web, tanto estático como dinámico, identifican y diferencian con colores código y texto, y vienen dotados de algunas potentes funciones para la gestión de archivos y carpetas, la búsqueda y sustitución masivas, la verificación y depuración del etiquetado, la comprobación de hipervínculos y otras referencias cruzadas, la conversión de caracteres especiales, entidades, etc. Muchos de ellos también vienen provistos de un visualizador propio o, al menos, permiten la interconexión con un navegador, así como de algún panel para visualizar y controlar la estructura de carpetas y archivos de un sitio web.

### 3.2. Aplicaciones ofimáticas y de autoedición

Hoy por hoy, prácticamente cualquier aplicación para el tratamiento de textos, la creación de presentaciones o la autoedición permite crear, procesar y visualizar archivos en los lenguajes de etiquetado más comunes, sobre todo HTML. Sin embargo, la mayoría de estas herramientas **reinterpretan** el etiquetado y, a menudo, añaden código propio, e incluso generan archivos nuevos, e inutilizan con ello el documento en cuestión para su re inserción en la estructura de carpetas del servidor web en el que esté alojado. Por consiguiente, su empleo en un proyecto de localización de sitios web es **totalmente desaconsejable**.

## Ventajas e inconvenientes de las aplicaciones ofimáticas y de autoedición

Ventajas	Inconvenientes
<p>Entorno gráfico WYSIWYG.</p> <p>Código oculto o protegido.</p> <p>Acceso a funciones avanzadas de tratamiento de textos (correctores ortográfico y gramatical, búsqueda y sustitución, macros, etc.).</p>	<p>Carencia de funciones de gestión de terminología y memorias de traducción y de traducción asistida en general.</p> <p>Modificación/inclusión/supresión de código e inclusión de firmas/huellas propias de la herramienta.</p> <p>A veces, la reinterpretación del código provoca su inutilización o acarrea la creación de nuevos archivos y carpetas.</p>

Por paradójico que parezca, más a menudo de lo que cabría esperar, muchos clientes — quizá no demasiado versados en las mejores prácticas del sector o temerosos de poner en peligro la integridad de su web— solicitan expresamente a sus proveedores de servicios de localización que se limiten a copiar los contenidos de la web en cuestión y pegarlos sin más en un documento de texto o en una hoja de cálculo para traducirlos sobrescribiéndolos.

Desde el punto de vista técnico, no es la solución más ortodoxa, eficiente ni elegante. Sin embargo, muchos clientes recurren a ella para garantizar la integridad de los componentes originales de la web: a veces, por desconfianza de la capacidad técnica del proveedor; otras, por puro desconocimiento. En cualquier caso, cuando esto sucede, resulta fundamental recalcarle al cliente la importancia de realizar un testeo en línea exhaustivo *a posteriori*, es decir, después de que los textos traducidos hayan sido reinsertados donde corresponda. Solo de este modo, se podrán detectar posibles errores y omisiones provocados por lo rudimentario del procedimiento.

### 3.3. Aplicaciones de diseño web

Con la excepción de aplicaciones profesionales que vengan provistas de funciones que eviten la reinterpretación del código y, en todo caso, permitan su depuración para garantizar la integridad del etiquetado original, la mayoría de las herramientas de creación y diseño web adolecen, en mayor o menor medida, de los mismos inconvenientes que las aplicaciones ofimáticas y de autoedición. Por lo demás, como es lógico, son herramientas específicamente diseñadas para la manipulación de los lenguajes empleados en la construcción de una web y, por tanto, ofrecen potentes prestaciones para la creación y el diseño de sitios web, como sucede con los editores de texto avanzados. Así pues, su empleo es, **en general, desaconsejable**, salvo si se trata de herramientas profesionales que garanticen la integridad del código fuente y se cuenta con conocimientos técnicos. En cualquier caso, al igual que ocurre en las anteriores categorías de herramientas, las de diseño web tampoco poseen las prestaciones propias de los gestores de memorias de traducción, para aumentar la productividad reciclando material ya traducido, garantizar la coherencia terminológica y fraseológica, etc.

## Ventajas e inconvenientes de las aplicaciones de diseño web

Ventajas	Inconvenientes
Entorno gráfico WYSIWYG. Código oculto o protegido. Potentes prestaciones de validación y depuración de código, control de carpetas y archivos, comprobación de hipervínculos, etc.	Carencia de funciones avanzadas de tratamiento de textos (correctores ortográfico y gramatical, búsqueda y sustitución, macros, etc.). Carencia de funciones de gestión de terminología y memorias de traducción y de traducción asistida en general. Modificación/inclusión/supresión de código e inclusión de firmas/huellas propias de la herramienta. A veces, la reinterpretación del código acarrea la creación de nuevos archivos y carpetas.

En determinadas situaciones, surgen tareas cuya resolución es inviable o engorrosa con otras herramientas (por ejemplo, la reordenación alfabética de enumeraciones, el rediseño de los campos de un formulario, etc.). En tales casos, puede resultar más sencillo y práctico —siempre y cuando la integridad del código fuente quede salvaguardada— efectuar estos retoques con una aplicación de diseño web.

### 3.4. Herramientas de traducción asistida

En la actualidad, cualquier sistema de gestión de memorias de traducción, tanto comercial como de software libre, de sobremesa o para trabajar en la nube, gestiona con solvencia la diferenciación de código informático y texto. Por tanto, su empleo resulta **muy recomendable**, puesto que el usuario puede sacarle el máximo partido a las prestaciones propias de una herramienta de este tipo (alineación, reaprovechamiento de repeticiones y coincidencias, búsquedas contextuales, gestión de terminología, análisis y procesamiento múltiple de archivos, pretraducción automatizada, manipulación de códigos y elementos transferibles, escritura predictiva, aseguramiento de la calidad, etc.). Según la herramienta elegida, la traducción se lleva a cabo en la propia interfaz autónoma de la aplicación, como si de cualquier otro formato se tratase, o existe un módulo o interfaz específicamente diseñados para la traducción de formatos menos comunes o más complejos.

## Ventajas e inconvenientes de los gestores de memorias de traducción

Ventajas	Inconvenientes
Las inherentes a este tipo de herramientas cuando las características del proyecto aconsejen su empleo (productividad, coherencia, etc.).	Imperfecciones y carencias de los filtros de importación/conversión de determinados formatos. A veces, problemas para la identificación de <i>scripts</i> . A veces, problemas de compatibilidad real entre herramientas diferentes (TMX). Sobre todo con XML o similares, disponibilidad de DTD o complejidad de definición de filtros <i>ad hoc</i> .

Cuando se manipulan formatos menos comunes o más complejos (como suele ser el caso de algunos de los empleados en webs dinámicas), puede resultar necesario definir manualmente filtros específicos para procesarlos garantizando la extracción íntegra de los textos traducibles y salvaguardando la integridad del código fuente.

### 3.5. Herramientas de traducción de páginas web

A las categorías anteriores, cabría añadir herramientas que han sido diseñadas con la localización de sitios web en mente y combinan prestaciones de los grupos anteriores. Aunque adolece de bastantes limitaciones, que se hacen patentes en proyectos de localización de sitios web de gran volumen o complejidad, el programa Catscradle, por ejemplo, destaca por su sencillez de manejo, cubre satisfactoriamente las necesidades de pequeños proyectos efectuados por profesionales autónomos –incluso de autotraducción–, y constituye una herramienta ideal para iniciarse en la localización de sitios web.

Traducción de archivo HTML con Catscradle

The screenshot shows the Catscradle software interface. The main window displays the translation of an HTML page. The original text is on the left, and the translated text is on the right. Below the text, there is a table with 16 rows, each representing a segment of the document. The table columns include 'Nº', 'Tipo', 'Texto origen', and 'Texto destino'. The original text is in Western Latin 1 (ISO-8859-1) and the destination text is in Unicode UTF-8. The interface also shows a menu bar, a toolbar, and a status bar at the bottom.

Nº	Tipo	Texto origen (358 palabras) Western Latin 1 (ISO-8859-1)	Texto destino (358 palabras) Unicode UTF-8
1	description	How a television screen produces a picture	How a television screen produces a picture
2	keywords	television, screen, tube, electron beam, electricity, light, primary colours, magnetism, electrons	television, screen, tube, electron beam, electricity, light, primary colours, magnetism, electrons
3	text	Television	Television
4	text	TELEVISION SCREENS	<a href="#">LAS PANTALLAS DE TELEVISIÓN</a>
5	text	The inside front of a television tube is coated with chemicals that react by emitting a spot of light when hit by a beam of electrons. Electrons are fired from an electron gun at the rear of the tube. The electron beam is deflected by magnetic coils placed between the gun and the front of the screen. The	The inside front of a television tube is coated with chemicals that react by emitting a spot of light when hit by a beam of electrons. Electrons are fired from an electron gun at the rear of the tube. The electron beam is deflected by magnetic coils placed between the gun and the front of the screen. The
6	text	magnetic field	magnetic field
7	text	generated by the coils is varied in a repeating pattern over time, so that the magnetically guided beam sweeps across the screen surface from left to right and top to bottom. The whole screen is traced by the beam in this way 50 times every second. This is so fast that to the human eye it appears as if the whole screen is lit up at once. By varying the intensity of the beam (based on the incoming television signal) as it traces the area of screen, the amount of light being emitted by the screen chemicals varies, tracing out a picture of light and dark.	generated by the coils is varied in a repeating pattern over time, so that the magnetically guided beam sweeps across the screen surface from left to right and top to bottom. The whole screen is traced by the beam in this way 50 times every second. This is so fast that to the human eye it appears as if the whole screen is lit up at once. By varying the intensity of the beam (based on the incoming television signal) as it traces the area of screen, the amount of light being emitted by the screen chemicals varies, tracing out a picture of light and dark.
8	text	COLOUR IMAGE	COLOUR IMAGE
9	img	RGB light diagram	RGB light diagram
10	text	To produce a colour image a television tube surface is coated in thousands of finely placed groups of chemical spots. Each group contains three spots of different chemicals designed to emit	To produce a colour image a television tube surface is coated in thousands of finely placed groups of chemical spots. Each group contains three spots of different chemicals designed to emit
11	text	red	red
12	text	green	green
13	text	and	and
14	text	blue	blue
15	text	light. There are three separate electron guns at the rear of the tube, each focused in such a way that the electron beams from one gun can only strike red emitting spots, one for green spots and one for blue. The red, green and blue spots in a group are so close to each other that they appear to be at the same point. As red, green and blue are the	light. There are three separate electron guns at the rear of the tube, each focused in such a way that the electron beams from one gun can only strike red emitting spots, one for green spots and one for blue. The red, green and blue spots in a group are so close to each other that they appear to be at the same point. As red, green and blue are the
16	text	primary colours of light	primary colours of light

La mencionada WebBudget se ha perfeccionado con los años, incluyendo paulatinamente prestaciones propias de los programas de teledescarga de webs o de las herramientas de traducción asistida, hasta convertirse en una completa aplicación con la que es posible acometer íntegramente la localización de una web estática de mediana complejidad, desde su descarga, análisis y presupuestación hasta la traducción de sus contenidos. Además, lleva integrado un completo catálogo de filtros con los que es posible procesar y traducir la mayoría de los formatos más comunes en cualquier web, tanto estática como dinámica.

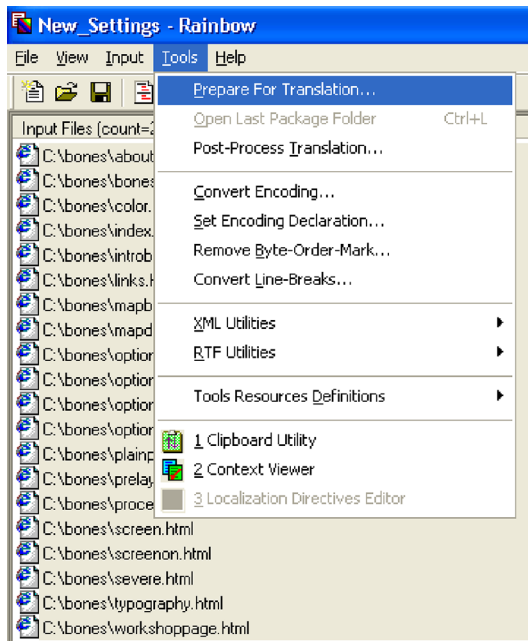
### 3.6. Otras herramientas

Como se decía al comienzo de este apartado, al igual que sucede en la localización de productos informáticos, en lugar de localizar archivos de etiquetado contenido web en su formato original con cualquiera de las herramientas mencionadas, cabe la posibilidad de extraerlos y convertirlos a un formato intermedio, como **RTF** (*rich text format*), **Opentag** o **XLIFF**, mediante algún sistema de exportación/etiquetado que permita su posterior reimportación o re inserción, preservando a toda costa su integridad. Puede ser aconsejable recurrir a tal estrategia, por ejemplo, para facilitar la traducción con otras herramientas o cuando no se cuenta con los conocimientos o programas necesarios para traducir el texto sin separarlo del código. Esta fue precisamente una de las estrategias más usadas en los primeros proyectos de localización de sitios web –por aquel entonces, literalmente «páginas»–, cuando ni las herramientas de traducción asistida ni las de localización eran capaces aún de procesar de manera eficaz este tipo de formatos, y se siguen utilizando hoy con profusión.

Este procedimiento semiautomático de exportación y reimportación es sencillo. Básicamente, consiste en **aislar** el código informático del documento en cuestión para diferenciarlo del texto y proteger su integridad mediante, por ejemplo, **estilos de carácter y de párrafo** personalizados, análogos a los predeterminados de cualquier procesador de textos. De ese modo, el traductor puede concentrarse en la traducción del texto, pasando por alto las etiquetas externas y reubicando, cuando sea necesario, las internas, las cuales reciben la consideración de elementos **transferibles** (*placeables*). Galibert (2004) describe el procedimiento con detalle y nos recuerda que existen otras posibilidades al alcance de profesionales autónomos y pequeños proveedores de servicios de traducción o localización, como el paquete de utilidades y filtros Okapi Framework o el complemento PlusTools para Wordfast.



Preparación de archivos HTML con Rainbow (una de las herramientas integradas en Okapi Framework)



Tras décadas de luces y sombras, el *boom* de la Red y el de la localización de sitios web también han traído de la mano la revitalización de la traducción automática y la popularización de algunos sistemas comerciales y servicios gratuitos. Amén del empleo de sistemas en línea para traducciones aproximadas (*gist translation*) de correspondencia electrónica, materiales de consulta, etc., tanto en la esfera profesional como en la personal, se observa un creciente interés por explotar su complementariedad con la traducción asistida y su integración en sistemas de gestión de contenidos (CMS), así como por el empleo de técnicas y herramientas de redacción controlada (*controlled authoring*) y de posesición, cuya definitiva implantación generalizada –mal que les pese a muchos– se prevé cercana.

## 4. El proceso y sus fases principales

En toda web existen dos niveles superpuestos en los que se estructura su contenido: por un lado, el **virtual** o visual, que constituye la organización –o mapa de navegación– que percibe el usuario cuando se desplaza por sus páginas y apartados, y cuya principal peculiaridad estriba en la hipertextualidad o posibilidad de moverse de manera no secuencial por su contenido; por otro, el **real** o subyacente, es decir, el conjunto organizado de carpetas, subcarpetas y archivos almacenados en el servidor en el que la web está hospedada.

En el nivel subyacente de estructuración real de una web, el conjunto de código informático (en lenguajes de etiquetado, de programación y de *script*) y contenido textual (y metatextual), gráfico y multimedia se almacena organizada en contenedores (archivos) que, como sucede en un producto informático, son básicamente de dos tipos: programas o archivos ejecutables por un procesador físico o lógico (integrados por instrucciones que «hacen cosas») y, sobre todo, documentos (que albergan el contenido). Aunque el contenido localizable suele hallarse en estos últimos, no es infrecuente que aparezca intercalado en el código de los primeros, como en el caso de los *scripts*. Además, existen otros archivos complementarios, como las hojas de estilo en cascada, CSS (*Cascade Style Sheet*), o las hojas de estilo XSL (*XML Stylesheet Language*), que controlan el formato, la apariencia y la uniformidad visual del contenido, pero no suelen contener elementos localizables. Ello no es óbice para que en determinadas circunstancias haya que hacer algún retoque en ellas para reajustar algún parámetro de formato (por ejemplo, tipo de letra, espaciado, márgenes, colores, etc.).

La diseminación y el nivel de atomización del contenido de una web en archivos diferentes son muy variables, y dependen en gran medida de los criterios de diseño adoptados en su concepción. Sin embargo, por pequeña y poco voluminosa que pueda parecer una web, por lo general se compone de decenas de archivos, que pueden llegar fácilmente a ser miles en el caso de webs de cierta entidad. A diferencia de lo que suele suceder en un producto informático (en el que un archivo compilado puede contener **en sí mismo**, como resultado de la compilación, numerosos recursos gráficos como cursores, iconos e imágenes de distinto tipo), en el caso del contenido web, los archivos escritos en lenguajes de etiquetado (HTML, por ejemplo) y de *script* (JavaScript, pongamos por caso) son a todos los efectos documentos de **texto sin formato** que, como tales, no permiten la inclusión de ningún elemento gráfico ni multimedia, sino tan solo de remisiones a los archivos que los alberguen.

Como ya se avanzaba en el módulo «El software», en los lenguajes de etiquetado (HTML, por ejemplo), el delimitador por excelencia es el corchete angular, también llamado antilambda o diple (< >, signo doble compuesto por el

«menor que», <, y el «mayor que», >), cuya principal peculiaridad estriba en albergar una secuencia de código informático con la que forma una **etiqueta** (en inglés, *tag*).

Las etiquetas pueden ser **sencillas** (como <P>, para introducir un párrafo nuevo) o **dobles** (como <B>texto</B>, para mostrar en negrita el texto delimitado), **internas** o **externas** (según acoten bloques de contenido completos o vayan intercaladas en ellos), y su finalidad atañe a la manera en que el contenido se presenta en un navegador, como en el caso de HTML (*HyperText Markup Language*), o a la estructura y el tipo de contenido de un documento, como en XML (*Extensible Markup Language*).

Los constructos empleados en HTML son, por orden jerárquico, los **elementos** (que pueden constar de una etiqueta sencilla o doble, con apertura y cierre), los **atributos** o **propiedades** (que modifican la función de un elemento, matizándola) y los **valores** que toman tales atributos (que pueden estar predeterminados o ser acuñados por el usuario mediante perfiles):

elementos → <etiqueta(s)> → atributos (propiedades) ="/valores"/ + comentarios

Además de estos constructos, pueden incluirse **comentarios**. En un documento HTML, los comentarios se atienen a la siguiente sintaxis, y su contenido no suele ser traducible:

```
<!-- Texto del comentario -->
```

Los elementos de un documento HTML pueden no incluir **ningún atributo**, como en:

```
<TITLE>Título del documento</TITLE>
```

Pueden contener **un atributo**, caso de:

```
Visite <A href="http://www.web.com/">nuestra web</A> si precisa más información.
```

O pueden llevar **varios atributos**, por ejemplo, en:

```
<IMG src="images/logo.png" alt="Logotipo de la empresa" border="0" align="absbottom">
```

#### Mayúsculas y minúsculas

En general, el empleo de mayúsculas o minúsculas es indiferente en el HTML (con la excepción de las entidades). Sin embargo, el XML es más estricto y establece la distinción entre ambas (*case sensitivity*).

Una de las reglas sintácticas básicas de estos lenguajes de etiquetado prescribe que los archivos constarán de una cabecera y un cuerpo, cuyo contenido irá delimitado respectivamente por las etiquetas dobles `<HEAD></HEAD>` y `<BODY></BODY>`, en el caso del HTML. Así, el **esqueleto básico de un documento HTML** sería el siguiente:

```
<HTML>
  <HEAD>
    <TITLE>Título del documento</TITLE>
  </HEAD>
  <BODY>
    [Contenido etiquetado del documento]
  </BODY>
  [...]
  [<SCRIPT>Instrucciones de script</SCRIPT>]
  [...]
</HTML>
```

El comienzo y el final del archivo van indicados por la doble etiqueta del elemento HTML, y lo primero que debe aparecer en un documento válido es la declaración de la versión de HTML empleada o, retomando nuestro símil musical, la clave que marcará la pauta para interpretar correctamente los códigos del pentagrama.

La cabecera, delimitada por las etiquetas `<HEAD>` y `</HEAD>`, suele contener información metatextual sobre el documento, y el **cuerpo**, acotado por las etiquetas `<BODY>` y `</BODY>`, alberga el grueso de su contenido con las etiquetas que corresponda en cada caso, cuya función será indicarle al navegador cómo ha de visualizarlo. Al traducir un archivo HTML –de manera análoga a cuando se traduce, por ejemplo, un libro–, puede tener sentido **comenzar por** el cuerpo y dejar para el final la cabecera. Dado que esta contiene información metatextual sobre el propio documento, se estará en mejores condiciones de abordar su traducción después de haber traducido íntegramente su contenido.

Es frecuente, aunque no obligatoria, la inclusión en cualquier punto del documento de **elementos dinámicos o interactivos** mediante instrucciones de programación en algún lenguaje de *script* (muy a menudo en JavaScript), las cuales van delimitadas por las etiquetas `<SCRIPT>` y `</SCRIPT>` y le ordenan al intérprete de este lenguaje con el que hoy ya cuenta cualquier navegador que ejecute una serie de acciones, como se explica más adelante.

A menudo –y es recomendable que así sea–, el formato global del contenido de uno o varios documentos HTML no se especifica mediante la inclusión de etiquetas en el cuerpo del propio archivo sino a través de una **hoja de estilos en cascada CSS** (*Cascade Style Sheet*), alojada en un archivo diferente. En tal caso, debe incluirse en la cabecera de cada documento HTML que utilice la

#### HTML5

La versión más reciente del HTML se publicó a finales de 2014. Introduce nuevos elementos y modifica o elimina otros. Supone una ostensible mejora de las versiones anteriores de este lenguaje, sobre todo, en lo que se refiere a estructuración, capacidad semántica y presentación del contenido, así como a interactividad y prestaciones multimedia, gracias su empleo combinado y mayor compatibilidad con tecnologías como CSS3, JavaScript o SVG (*Scalar Vector Graphics*), entre otras.

hoja de estilos en cuestión una referencia al archivo CSS (por ejemplo, mediante el elemento LINK), de manera que el navegador muestre el contenido correctamente.

```
<LINK rel="stylesheet" type="text/css" href="css/styles.css"></LINK>
```

Hemos mencionado algunos de los fundamentos básicos de la **gramática** que emplean los lenguajes de etiquetado para estructurar el contenido web, y que en su día se sistematizó en el *Standard Generalised Markup Language* (SGML), conjunto de especificaciones en el que se basan lenguajes como HTML, DHTML, XML, XHTML, entre otros. De todos ellos, los dos más empleados en la actualidad son HTML y XML. Su parentesco radica en atenerse a las normas dictadas por el SGML; la principal diferencia es consecuencia de su finalidad, ya mencionada. No en vano, el HTML es un lenguaje con un léxico finito, puesto que consta de un número limitado de etiquetas, mientras que el XML y sus derivados o aplicaciones le permiten al usuario acuñar sus propias etiquetas sin más limitación que las impuestas por la gramática SGML. Todo documento basado en ella debe explicitar su significado y función mediante una DTD (*Document Type Definition*) o un esquema (*schema*), que no es más que un **vocabulario** específico en el que se explican la estructura y las etiquetas del documento al que está asociada.

Como ya se avanzaba en el módulo «Conceptos básicos» de este curso, cabría formular **tres reglas básicas**, heredadas de las prácticas que convencionalmente se adoptan en la localización de software y que cabe aplicar también a los lenguajes de etiquetado. En ellos, el delimitador (la <etiqueta>) sí alberga contenido, tanto textual como código informático, y:

- 1) Por lo general, el texto localizable suele ir intercalado entre etiquetas (<etiqueta>texto localizable</etiqueta>).
- 2) Por lo general, el contenido de las etiquetas no suele ser localizable, puesto que se trata casi siempre de código informático.
- 3) Las dos reglas anteriores presentan algunas excepciones, pero son pocas y conocidas, y casi siempre se producen en etiquetas sencillas.

Estos últimos **casos especiales** a los que nos referimos pueden constituir una excepción, bien porque exista texto o código localizable dentro de una etiqueta, o bien por otras circunstancias especiales como, por ejemplo, que el texto no aparezca como parte del contenido inmediatamente visible mostrado por un navegador, como sucede con el elemento META. Con la salvedad de tales excepciones, el resto del texto traducible no debería entrañar ninguna dificul-

### W3C

El organismo encargado de desarrollar y normalizar las tecnologías HTML, DHTML, XML y XHTML a nivel mundial es el World Wide Web Consortium ([www.w3c.org](http://www.w3c.org)).

### DTD

No disponer de la DTD o del esquema específicos de un determinado archivo XML puede plantearles problemas a algunas herramientas de traducción asistida o de localización que, como el navegador, deben disponer de ellos para poder **interpretar** correctamente el contenido del documento XML. Véase a este respecto la web del proyecto colaborativo Schema.org.

tad si se aplican sistemáticamente las reglas anteriores, como, de hecho, hacen las herramientas que suelen emplearse para localizar archivos de etiquetado y que detallamos en los apartados que siguen.

#### 4.1. Los elementos textuales

Para traducir los elementos textuales de un documento HTML, sirva como ejemplo de aplicación de las reglas anteriores el elemento `TITLE`, que se incluye en la cabecera de un documento HTML y contiene su título. Algunas de sus peculiaridades son las siguientes:

- Es un elemento obligatorio y debe aparecer una sola vez en la cabecera del documento.
- Su contenido no aparece inmediatamente visible en ningún lugar de la página, sino en la **barra de título** de la ventana del navegador, seguido del nombre de este y separado de él por un guión (Título del documento - Navegador), como en cualquier otra aplicación o en sus pestañas.
- Lo consultan y usan los buscadores para indexar el documento y lo muestran entre sus resultados en la SERP (Search Engine Results Page), factores que deben tenerse muy en cuenta al traducirlo, puesto que puede influir en el posicionamiento, o SEO (Search Engine Optimization).
- El título de un documento HTML y su traducción han de ser todo lo descriptivos del contenido de la página que se pueda, y es recomendable – como en el caso del elemento `META`– adoptar una estrategia de traducción más funcional que semántica, o sea, que atienda al **cometido** del texto (original) más que a su **significado**.

Los demás elementos textuales siguen las mismas pautas, y su traducción no debería plantear ninguna dificultad con alguna de las herramientas mencionadas en apartados anteriores, sea cual sea la estrategia elegida. Estas cumplen el ya citado doble cometido de separar el código informático del texto susceptible de ser traducido, y preservar la integridad de aquel.

Por ejemplo, en la siguiente ilustración, un documento HTML se ha etiquetado previamente mediante los estilos «tw4winExternal» (de color gris) y «tw4winInternal» (de color rojo), que respectivamente protegen las etiquetas externas e internas, de modo que no puedan modificarse de manera accidental. Una vez etiquetado el archivo, puede procederse a su traducción con cualquier herramienta de traducción asistida; incluso con aquellas que no tienen interfaz propia y funcionan a través de la propia del procesador de textos (como, por ejemplo, Anaphraseus, Wordfast Classic o el ya obsoleto Trados Translator's Workbench, antecesor del actual SDL Trados Studio).

Traducción de documento HTML con Wordfast Classic, después de haberlo etiquetado mediante estilos con PlusTools

```
<td valign="top" width=20>
<td valign="top" width=220>
<font face="Verdana" size=-2 color=navy>
[
{0}
<b>Serveis</b> ]
</td>
<b>Servicios</b> ]
</td>
<br>
Traducció de software, pàgines web i documentació<br>
```

En la siguiente figura, se ilustra cómo se puede traducir un documento HTML en un editor que reconoce las etiquetas y protege su integridad. Las etiquetas reciben el tratamiento de **elementos transferibles**, de manera que se pueden reproducir en los segmentos traducidos.

Traducción de documento HTML con TagEditor, predecesor de SDL Trados Studio

```
<td valign="top" width=20>
<td valign="top" width=220>
<font face="Verdana" size=-2 color=navy>
<b> Serveis </b>
<b> Servicios </b>
<br>  Traducció de software, pàgines web i documentació <br>
```

## 4.2. Los elementos metatextuales

Ni todo lo que el cibernauta ve en un sitio web se localiza, ni todo aquello que se localiza resulta visible para quien lo visita. Tal es el caso de la información metatextual que optativamente puede contener un sitio web y que, por lo general, se articula en su código mediante el elemento `META`. A continuación, se detallan su sintaxis y algunas particularidades.

Sintaxis del elemento `META`

Atributos	Sintaxis	Descripción
name= content= scheme= http-equiv=	<META atributo="valor no localizable" [...] content="valor localizable">	Metainformación sobre el documento.

- No es obligatorio, pero sí frecuente –y muy recomendable–, que este elemento se incluya en la cabecera de un documento HTML con atributos y valores diferentes.
- Su contenido no aparece visible en ningún lugar de la página ni de la ventana del navegador, pero lo consultan los buscadores y otros sistemas para indexar la página, factor que debe tenerse muy en cuenta al traducir su contenido, puesto que, en algunos casos, también puede influir en el **posicionamiento** de la web en los resultados de los buscadores.
- No tiene etiqueta de cierre y, por tanto, el texto traducible va dentro de la única etiqueta <META>, entrecomillado e inmediatamente precedido del atributo CONTENT.
- Los valores que toman los atributos pueden ser elegidos por el usuario o tomados de un perfil predefinido. En todo caso, nunca se traducen los valores del atributo NAME.

### Posicionamiento (SEO)

Singular importancia reviste el empleo —meditado, sistemático y no abusivo, para no caer en el denominado *black-hat SEO*— de palabras clave (*keywords*) tanto en el título y la cabecera de un documento HTML como, sobre todo, en lugares estratégicos de su cuerpo (como encabezamientos, hipervínculos, texto alternativo de imágenes y otros elementos que se describen en los apartados que siguen). Todos ellos —junto con otros que no se mencionan por su complejidad técnica— contribuyen coligadamente al posicionamiento u optimización en los buscadores, o SEO (Search Engine Optimization).

### Contenido metatextual

Como recuerdan, entre otros, Galibert (2004) o el propio Google –hoy, buscador por antonomasia–, la traducción del contenido textual del elemento TITLE o de algunos subtipos del elemento META (como, por ejemplo, aquellos cuyo atributo NAME= contiene el valor «Description» o «Keywords») que consultan los buscadores para indexar las páginas que los contienen debe hacerse atendiendo no tanto a la mera transferencia de significados recogidos en el original, sino pensando más en la función que cumple este contenido **metatextual**. Esto implica, por ejemplo, que se empleen en la traducción términos susceptibles de ser introducidos en un buscador por alguien que pretenda llegar a una web o que se reflejen deliberadamente en la versión traducida sinónimos probables o, incluso, variantes ortográficas o erratas frecuentes, con el objetivo de aumentar las probabilidades de que la web en cuestión aparezca entre los primeros resultados de la búsqueda y el navegante acabe visitándola.

### Ejemplo

A continuación, se ilustran con un ejemplo real (tomado de la versión en alemán del sitio web [www.trans-it.com](http://www.trans-it.com)) algunos subtipos de elemento <META> que pueden contener, como valor del atributo CONTENT, texto traducible o código de etiquetado susceptible de ser adaptado. Como puede observarse, en algunos casos el valor del atributo CONTENT se traduce (en verde), mientras que en otros se trata de código informático que permanece invariable (en rojo) o que podría ser preciso adaptar (en naranja), como el *charset* juego de caracteres elegido para codificar la página.

#### Ejemplos del elemento META

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; CHARSET=iso-8859-1">
<META NAME="Title" CONTENT="TRANS-IT - Was wir Ihnen bieten">
<META NAME="author" CONTENT="TRANS-IT: Birgit Schöbitz, Birgit Hofmann, Dzifa
Vode">
<META NAME="copyright" CONTENT="TRANS-IT Fachübersetzungen Birgit Schöbitz">
<META NAME="keywords" CONTENT="Übersetzung, Übersetzungen, Fachübersetzung,
Bedienungsanleitung, technische Dokumentation, Prospekt, Trans-It,
Fremdsprachen, Übersetzer, Englisch, Französisch, Italienisch, Spanisch,
Korrekturlesen, Niederländisch, Deutsch, Dolmetschen, Website, Handbuch">
<META NAME="description" CONTENT="Sprachmittlerservice: Bei uns übersetzen
grundsätzlich nur Muttersprachler mit einschlägiger Erfahrung im jeweiligen
Fachgebiet: Technik, Wissenschaft, Recht, Wirtschaft, Politik und vieles
mehr.">
<META NAME="audience" CONTENT="all">
<META NAME="robots" CONTENT="INDEX, FOLLOW">
<META NAME="revisit-after" CONTENT="20 days">
```



### 4.3. Los elementos interactivos

Dentro de un sitio web, el elemento más característico lo constituyen, sin duda, sus hipervínculos, que se articulan en el código mediante el elemento `A`, el cual se compone de la doble etiqueta `<A></A>` y contiene, además del texto visible que invita al usuario a pulsarlo, el **destino** del hipervínculo (lugar o documento del mismo sitio o de otra web, aplicación de correo, etc.).

Ejemplos del elemento `A`

fragmento de web visualizado en un navegador	código HTML del fragmento
<p>If you have any questions about our localization services, references, profiles, etc, or you would like to receive a free sample translation, or a detailed project quotation just for your guidance, please <a href="#">send us a message</a> and we will get back to you immediately.</p> <p>We are based in Barcelona, Spain, with production partners in more than ten countries, and representatives in the USA and France.</p> <p>Professional translators, engineers and specialized localizers who might be interested in working with us are invited to kindly <a href="#">fill in this form</a>.</p>	<pre>If you have any questions about our localization services, references, profiles, etc, or you would like to receive a free sample translation, or a detailed project quotation just for your guidance, please &lt;a href="mailto:info@localversion.com" "&gt;send us a message&lt;/a&gt; and we will get back to you immediately.&lt;p&gt; [... ] Professional translators, engineers and specialized localizers who might be interested in working with us are invited to kindly &lt;a href = tform.htm&gt;fill in this form&lt;/a&gt;.&lt;p&gt;</pre>

A continuación, se resumen la sintaxis y también algunas particularidades del elemento `A`.

Sintaxis del elemento `A`

Atributos	Sintaxis	Descripción
name= href= scheme= http- equiv= hreflang= type= rel= rev= charset=	<pre>&lt;A atributo="valor no localizable" [... ] href="valor localiza- ble"&gt; texto localizable &lt;/A&gt;</pre>	Ancla ( <i>anchor</i> ) o hipervínculo a un recurso.

- No es obligatorio, pero sí frecuente, que el elemento `A` se incluya varias veces en el cuerpo de un documento HTML con atributos y valores distintos.
- Su contenido sí aparece visible en algún lugar de la página como hipervínculo, con los atributos de formato que le correspondan según la configuración del navegador, u otros parámetros establecidos en el etiquetado del documento (habitualmente, color azul y subrayado antes de visitarlo, o color violeta cuando ya se ha visitado).
- El texto del hipervínculo (*anchor text*) contribuye no solo al posicionamiento de una web sino también a su grado de accesibilidad. Por tanto el empleo de *anclas* como «aquí», «más» o similares repercute negativamente en ambos factores, motivo por el cual conviene adoptar —también aquí— una estrategia de traducción/localización funcional más que una estricta-

mente semántica; a menudo, muy a pesar de lo que diga el texto original (por ejemplo, *Click [here](#)*).

- En realidad, el elemento A no constituye una excepción a las reglas que hemos enunciado, puesto que el texto localizable va fuera de las etiquetas y está perfectamente delimitado por ellas. Basta con traducirlo cuando proceda. Sin embargo, en algunos casos puede ser necesario retocar el valor del atributo `HREF` (que representa la ubicación de un recurso web mediante un URI o un URL), o reemplazarlo por el que se desee en la versión localizada del documento. Como se ilustra en los ejemplos, un elemento A podría remitir a la web o una de sus partes, a un documento, a una dirección de correo o a un servidor FTP de la sede central de una multinacional, mientras que en la versión localizada, el hipervínculo conectaría con los respectivos recursos locales de la filial del país de destino.

### URL

Un URL (*Uniform Resource Locator*), comúnmente interpretado como la dirección de una web, es un tipo de URI (*Uniform Resource Identifier*) y se utiliza para identificar y localizar inequívocamente un recurso, o esquema, al que puede accederse a través de alguno de los protocolos propios de internet como, por ejemplo, una web, una dirección de correo electrónico, un servidor de FTP, etc.

Además de los hipervínculos, la localización de sitios web puede suponer también la de archivos escritos en lenguajes de *script*. Tanto si las instrucciones de un *script* han sido externalizadas y se alojan en archivos independientes, como si van entreveradas en el etiquetado, las estrategias que se adoptan son similares a las empleadas en localización de software, que se describen en otros módulos de este curso. No obstante, dado que los lenguajes de *script* –en particular, JavaScript– se emplean con profusión incluso en webs semiestáticas en las que toda la interactividad se produce en el cliente y no en el servidor, a continuación se recoge un ejemplo que ilustra la problemática que su localización plantea.

En el siguiente ejemplo real (tomado de <http://accurapid.com/journal/form.html>), se presenta un formulario en el que la interactividad con el usuario se articula de dos maneras. Por un lado, el elemento `FORM` de HTML le presenta un formulario y le ofrece la posibilidad de interactuar con él rellenándolo, borrando los datos introducidos con el botón *clear form* o enviándolo con el botón *send*. Por otro, un breve *script* escrito en JavaScript incluido en el etiquetado HTML controla que el formulario se cumplimente íntegramente antes de ser enviado y, de no ser así, muestra un mensaje de error, que insta al usuario a que rellene los campos vacíos.

### Direcciones de correo electrónico

Es frecuente que el elemento A se emplee, además de para incluir en un documento un hipervínculo a otra web, para crear hipervínculos con direcciones de correo electrónico. En tal caso, cuando el usuario hace clic en el hipervínculo, se activa automáticamente la aplicación de correo que se tenga instalada para enviar un mensaje a la dirección hipervinculada. En tal caso, a veces es preciso adaptar la dirección de correo reemplazándola por la que corresponda en la versión localizada, medida que conviene consultar al cliente para cerciorarse de cuál es la dirección disponible o deseada en la web localizada.



Como se observa en el código del archivo *form.html*, reproducido íntegramente a continuación, el *script* va delimitado por la doble etiqueta `<SCRIPT>` y `</SCRIPT>`, y se ha incluido en la cabecera del archivo (delimitada por el par de etiquetas `<HEAD>...</HEAD>`), mientras que el formulario (acotado por las etiquetas `<FORM>` y `</FORM>`) constituye el cuerpo del documento HTML (delimitado por el doble elemento `<BODY>...</BODY>`). La localización del *script* se reduciría a traducir el texto del mensaje de error (por ejemplo, con un editor de textos avanzado), que aparece con doble entrecomillado como valor de la variable *msg*. Para la localización del etiquetado HTML, incluido el del formulario, pueden seguirse las pautas explicadas anteriormente para traducir elementos textuales.

En este tipo de formularios, tan comunes en cualquier web, es frecuente que se plantee la necesidad de retocar incluso su estructura y diseño para adaptarlos al mercado de destino. Tal es el caso, por ejemplo, de las listas desplegables de elementos ordenados alfabéticamente, o de los campos u opciones que puedan tener o no sentido (como el destinado al segundo apellido en España) o cuyas dimensiones no resulten adecuadas en la versión localizada.

Elementos SCRIPT y FORM incluidos en un archivo HTML

```

1 <html>
2 <head>
3 <title>TJ Feedback</title>
4 <script LANGUAGE="JavaScript">
5 function Validate(form)
6 {
7     var msg = "You must enter your name, e-mail and location.";
8     var realname = document.forms[0].realname.value;
9     var email = document.forms[0].email.value;
10    var location = document.forms[0].location.value;
11    // check for any space characters that may be input
12    if (realname == "" || realname == null || email == "" ||
13        email == null || location == "" || location == null)
14    {
15        alert (msg);
16        return false;
17    }
18 }
19 </script>
20 </head>
21
22 <body>
23 <form method="POST" action="http://www accurapid.com/cgi bin/formmail.php">
24 <pre>
25 your name: <input name="realname" size=25>
26 e-mail address: <input name="email" size=25>
27 your location: <input name="location" size=25>
28 your message:
29 <textarea wrap=soft name="Feedback" rows=4 cols=50></textarea>
30 </pre>
31 <p>
32 <input onClick="return Validate(this.form)" type=submit value="send">
33 <input type=reset value="clear form">
34 <input type=hidden name="recipient" value="gbokor@accurapid.com">
35 <input type=hidden name="subject" value="TJ Feedback">
36 <input type=hidden name="redirect" value="thanks.html">
37 </form>
38 </body>
39 </html>

```

En el ejemplo anterior, el código del *script* va intercalado en el etiquetado HTML. Sin embargo, también es frecuente que se aloje en un archivo independiente (con extensión JS en el caso de JavaScript), de modo que en el documento HTML se incluye tan solo una referencia al archivo JS (precedida del atributo SRC del elemento `<SCRIPT>`) para ejecutar sus instrucciones:

```
<script src="miscript.js" type="text/javascript"></script>
```

Tanto si el código del *script* se incluye en el etiquetado HTML, como si se almacena en un archivo por separado, se trata de archivos de texto sin formato y, para su localización, puede emplearse cualquier herramienta que permita procesar este formato.

#### 4.4. Los elementos gráficos

La localización de contenido gráfico en una web supone una complicación más en los flujos de trabajo y la gestión de un proyecto de localización, pero no difiere sustancialmente, en cuanto a las técnicas y herramientas que requiere, de la localización de software, como se explica en otros módulos del curso.

En el etiquetado HTML, el elemento que permite incluir imágenes es la etiqueta sencilla `<IMG>`, cuyas principales características se resumen a continuación, junto con su sintaxis.

Sintaxis del elemento IMG

Atributos	Sintaxis	Descripción
src= longdesc= name= [alt=] [href=]	<IMG atributo="valor no localizable" [...] src="valor localizable" alt="texto localizable">	Imagen que se muestra en un documento.

#### Accesibilidad web

Hoy día el texto alternativo de las imágenes es uno de los elementos fundamentales en la accesibilidad de una web. Los lectores de pantalla que emplean las personas con alguna disfunción visual convierten a audio el texto que aparece como valor del atributo ALT. Por eso, debe ponerse especial cuidado en su correcta detección y traducción atendiendo —una vez más— a criterios funcionales más que estrictamente traductológicos.

- No es obligatorio, pero sí muy frecuente, que el elemento `IMG` se incluya en el cuerpo varias veces con atributos y valores diferentes.
- Su contenido textual sí aparece visible, pero no como parte de la página, sino como texto alternativo (en el código, este texto va precedido del atributo `ALT`), que se visualiza en un pequeño recuadro al situar el cursor sobre la imagen en cuestión. Su finalidad es informar al usuario sobre la imagen, en caso de que esta no se pueda visualizar por algún motivo.
- El atributo `ALT` no es específico del elemento `IMG` (también puede emplearse con los elementos `AREA`, `APPLET` o `INPUT`), aunque su problemática es la misma.
- El atributo `SRC` sirve para especificar la ubicación de la imagen que ha de mostrarse en la página. La problemática de su localización es muy similar a la del atributo `HREF` del elemento `A`, y puede ser necesario adaptar su valor cuando, por ejemplo, la imagen localizada tenga un nombre y/o una ubicación diferentes a los de la imagen original.

- Es frecuente que una imagen se hipervincule anidando el elemento `IMG` correspondiente entre las etiquetas `<A></A>`, como sucede con frecuencia para representar los botones o menús de navegación de un sitio web.

En los siguientes ejemplos reales (versiones en catalán y castellano de [www.localversion.com](http://www.localversion.com)), aparecen algunos elementos `IMG` en los que se traduce el valor del atributo `ALT` (en verde) y se ajusta el del atributo `SRC` (en naranja), puesto que las imágenes a las que se hace referencia desde el documento están almacenadas en archivos diferentes para cada lengua. Todos los elementos `IMG`, salvo el último (que corresponde a una barra no localizable, común a todas las lenguas), van anidados en sendos elementos `A`, de manera que las imágenes quedan hipervinculadas a los documentos de destino especificados en cada caso.

- El nombre de los archivos de una web también contribuye al SEO, y las imágenes no son una excepción. Este motivo sería una razón de peso para justificar la traducción de los nombres de los archivos de una web. Sin embargo, hacerlo tiene la contrapartida de multiplicar los retoques del etiquetado (por ejemplo, los atributos `HREF` de los hipervínculos o los `SRC` de las imágenes) y, con ello, complicar el mantenimiento de una web e incrementar el coste de su localización.

Imágenes y etiquetas `<IMG>` de archivo HTML en catalán

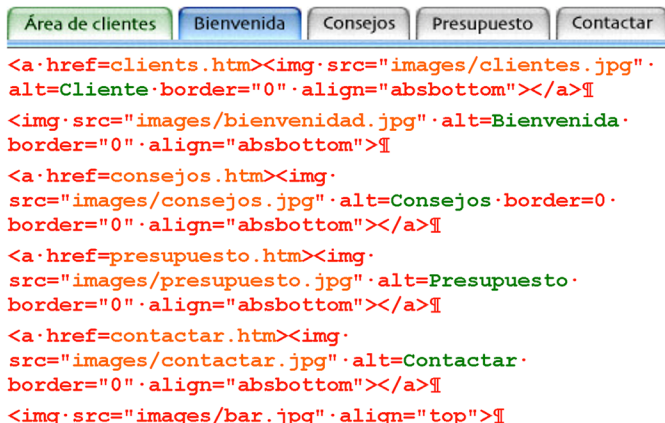
```






<a href=clients.htm></a>¶
¶
<a href=consells.htm></a>¶
<a href=pressupost.htm></a>¶
<a href=contacte.htm></a>¶
¶
```

Imágenes y etiquetas <IMG> de archivo HTML en castellano



Como sucede en los ejemplos anteriores, sigue constituyendo la norma *de facto*, más que la excepción, el empleo de **imágenes con texto incrustado** en los menús, botones y otros elementos gráficos de una web, como *banners* o ventanas emergentes, cuando lo recomendable sería externalizar el contenido textual para facilitar su localización. La problemática que plantea el necesario tratamiento al que han de someterse tales imágenes para poder localizarlas se agrava, porque pocas veces se conservan o se ponen a disposición del proveedor los archivos en los formatos nativos en cuestión (por ejemplo, en formato PSD), en los cuales el texto está físicamente separado del resto del contenido en capas (*layers*), y localizarlo resultaría sumamente sencillo. Informar al cliente de estos extremos, y educarlo en la adopción de buenas prácticas en este sentido, constituye una medida esencial para simplificar el proceso de localización de contenido gráfico y reducir así, drásticamente, costes y plazos de entrega.

En lo que atañe a los formatos que se emplean convencionalmente, lo habitual en la Red es emplear formatos comprimidos o más ligeros (en detrimento de la resolución de las imágenes, que en internet puede resultar accesoria), como GIF (*Graphic Interchange Format*) para gráficos, dibujos, diagramas, organigramas, sencillas animaciones por fotogramas, etc., o JP(E)G (*Joint Photographic Experts Group*) para imágenes fotográficas. Los dos han desplazado a otros formatos tradicionales que aún se usan profusamente en proyectos de localización de productos informáticos, como BMP (*BitMaP*). Recientemente está proliferando el uso del PNG (*Portable Network Graphic*) y, aunque aún se antoja remota, también parece previsible –y deseable a todas luces– la implantación paulatina de formatos vectoriales basados en XML, como SVG (*Scalable Vector Graphics*), tanto por su menor tamaño como por las evidentes ventajas de su tratamiento y localización. Estos aspectos y formatos se abordan con profusión de detalles en el módulo "Las imágenes y el contenido gráfico".

Las **estrategias** y **herramientas** empleadas para el tratamiento de contenido gráfico no difieren prácticamente de las usadas en un proyecto de localización de productos informáticos, o incluso en un proyecto de traducción convencional que incluya la traducción de elementos de este tipo, y lo habitual es emplear aplicaciones profesionales de diseño y edición de imágenes. Además,

en los últimos años han surgido en este ámbito herramientas específicamente diseñadas para ser utilizadas en proyectos de localización. Bien conocidas entre los profesionales del sector son utilidades como Image Browser o Globalisation Image Assistant, para agilizar el procesamiento y la presupuestación de grandes cantidades de archivos gráficos en varios formatos, o los filtros incluidos en el paquete Okapi Framework, para semiautomatizar la extracción y re inserción de las capas de texto de imágenes creadas con algunas aplicaciones de diseño gráfico.

#### 4.5. Otros elementos localizables

Como ya se ha mencionado, cuando se localiza un sitio web –o idealmente, cuando se diseña–, una de las tareas preliminares consiste en idear la manera en la que se organizarán jerárquicamente sus contenidos, previendo las implicaciones de su localización o de posibles ampliaciones y reestructuraciones, para minimizar su repercusión.

Así, existen webs en las que el idioma original constituye el eje principal, y los contenidos localizados a otros idiomas se estructuran en subcarpetas dependientes de él. Otra organización posible consiste en ubicar todos los contenidos de manera adyacente al mismo nivel, especialmente en webs de reducidas dimensiones. En ocasiones, se opta por racionalizar la organización de los contenidos y almacenar los archivos comunes (por ejemplo, hojas de estilo, imágenes no localizables y otros recursos compartidos por todas las versiones, como *scripts*) en una o varias carpetas centralizadas, a las que se accede desde los archivos de las distintas versiones que comparten dichos recursos. También es frecuente que las grandes compañías e instituciones opten por construir webs autónomas, vinculadas en mayor o menor medida y con más o menos contenido compartido, pero organizadas por lenguas, comunidades de destino, países o incluso regiones.

La previsión resulta fundamental también en este aspecto, puesto que no tomar inicialmente en consideración que una web vaya a acabar localizándose provoca que la organización de sus contenidos no prevea la inclusión de las versiones localizadas, y esto puede suponer la completa reestructuración del árbol de archivos y carpetas, amén de complicar su mantenimiento.

Cuando se acomete la localización íntegra de un sitio web, caben básicamente dos posibilidades: que este ya sea multilingüe, o que aún sea monolingüe. En el primer caso, lo más práctico es imitar la estructura de archivos y carpetas que ya exista, y acoplar la nueva versión localizada al conjunto. En el segundo, se plantean las posibilidades que se acaban de apuntar, y lo lógico es plantearle al cliente y negociar con él cuál se considera la estructuración idónea.

Dejando a un lado los aspectos estrictamente técnicos, al localizar un sitio web no han de olvidarse las implicaciones culturales que esto plantea. Desde una óptica eminentemente intercultural, y como consecuencia también de

una deficiente o nula internacionalización previa, sigue siendo habitual incluir elementos textuales y metatextuales, gráficos, imágenes o audio y vídeo que guardan una estrecha vinculación con la cultura de origen y que, por tanto, es preciso modificar, reemplazar o incluso eliminar en las versiones localizadas. En este sentido, no ha de perderse de vista que tanto una internacionalización **aseptizante** llevada al extremo, como una localización excesivamente superficial, pueden provocar una **aculturación** de los contenidos, que acabe desprovendolos de identidad y valores; y, a menudo, de autenticidad, naturalidad, gancho o chispa.



## Resumen

Como ya se avanzaba en el módulo «La localización: conceptos básicos» de este curso, parece que hay algunas diferencias notables entre localización de productos informáticos y de sitios web, que repercuten de manera decisiva en los formatos, las estrategias, las herramientas y los procedimientos que se emplean habitualmente en las dos modalidades; así como otra diferenciación también apuntada: la que cabría trazar entre sitios web estáticos y dinámicos, atendiendo a la relación entre el servidor en el que se aloja una web y los clientes que la visitan.

Por lo demás, la localización de sitios y contenido web constituye un mercado que mueve cantidades ingentes de contenido susceptible de ser localizado, y no solo a las lenguas tradicionalmente receptoras en localización de software. A ello han contribuido, sin duda, la consolidación de las redes sociales y de nuevos formatos y canales de comunicación e interacción a través de la Red. Basta pensar en cómo son las webs y los contenidos que consultamos a diario y en cómo y en qué medida están localizados. Las oportunidades laborales en el ámbito de la localización de sitios web son, si cabe, mayores, ya que un profesional autónomo o una modesta empresa que cuenten con las competencias y herramientas necesarias pueden plantearse acometer el proceso en todas sus fases y ofrecer un servicio integral, al menos, en el caso de sitios web estáticos o semiestáticos de mediana entidad y complejidad.

Paradójicamente, el volumen de contenido «corporativo» (descripción o historia de la entidad, datos de contacto, etc.), a menudo de naturaleza estática, de la web de cualquier emprendedor o PYME ha mermado manifiestamente en los últimos años; casi tanto como también lo han hecho los patrones y hábitos de lectura, los dispositivos con los que accedemos a la información, y las preferencias estéticas y tendencias de diseño de los tiempos que corren.



## Bibliografía

**Esselink, B.** (2000). *A Practical Guide to Localization*. Ámsterdam/Filadelfia: John Benjamins.

**Esselink, B.** (2001). «From Software Localization to Web Globalization –and back». *Multilingual Webmaster*. ForeignExchange Translations [documento en línea]. <[http://www.multilingualwebmaster.com/library/software\\_localization.html](http://www.multilingualwebmaster.com/library/software_localization.html)>

**Galibert, S.** (2004). «Translating Web Sites. HTML for Translators». *YourTranslations.com* [documento en línea].<[http://www.your-translations.com/html-for-translator\\_1.php](http://www.your-translations.com/html-for-translator_1.php)>

**Grupo Tradumàtica (ed.)** (2002). «La localización / La localització». *Revista Tradumàtica* (núm. 1) [documento en línea]. <[http://www.fti.uab.es/tradumatica/revista/index\\_01.htm](http://www.fti.uab.es/tradumatica/revista/index_01.htm)>

**Jiménez Crespo, M. Á.** (2013). *Translation and Web Localization*. Londres/NuevaYork: Routledge.

