

Lenguajes de marcas: fáciles de escribir, bonitos de leer

Quico Saval Vicente

Última modificación: lunes, 03 de diciembre de 2012

Índice

- [1. Consideraciones previas](#)
 - [1.1. Introducción](#)
 - [1.2. Conocimientos previos](#)
 - [1.3. La actitud para el aprendizaje](#)
- [2. La edición de textos](#)
 - [2.1. Introducción](#)
 - [2.2. La edición de textos tradicional](#)
 - [2.3. Los lenguajes de marcas](#)
 - [2.3.1. Internet se escribe con un lenguaje de marcas](#)
 - [2.3.2. Otros lenguajes de marcas](#)
- [3. Los elementos de HTML](#)
- [4. Las herramientas y el proceso de trabajo](#)
 - [4.1. Introducción](#)
 - [4.2. Explicación para Windows](#)
 - [4.2.1. Descarga e instalación](#)
 - [4.2.2. Uso de EdSharp](#)
 - [4.2.3. Configurando EdSharp](#)
 - [4.2.4. El proceso de conversión](#)
 - [4.3. Explicación para otros sistemas operativos](#)
- [5. Markdown](#)
 - [5.1. La sintaxis](#)
 - [5.1.1. La estructura del documento](#)
 - [5.1.2. Capítulos, apartados y subapartados](#)
 - [5.1.3. Párrafos y saltos de línea](#)
 - [5.1.4. Las listas](#)
 - [5.1.5. Enlaces](#)
 - [5.1.6. Imágenes](#)
 - [5.1.7. Citas](#)
 - [5.1.8. Enfatizando textos](#)
 - [5.1.9. Código sin procesar](#)
 - [5.1.10. Creando el índice](#)
 - [5.1.11. Caracteres especiales](#)
 - [5.1.12. Otros elementos](#)
 - [5.2. Opciones de conversión](#)
- [6. Txt2tags](#)
 - [6.1. La sintaxis](#)
 - [6.1.1. La estructura del documento](#)
 - [6.1.2. Capítulos, apartados y subapartados](#)
 - [6.1.3. Párrafos](#)
 - [6.1.4. Listas](#)

- [6.1.5. Enlaces](#)
- [6.1.6. Imágenes](#)
- [6.1.7. Destacando textos](#)
- [6.1.8. Citas](#)
- [6.1.9. Tablas](#)
- [6.1.10. Líneas horizontales](#)
- [6.1.11. Código sin procesar](#)
- [6.1.12. Creando el índice](#)
- [6.1.13. Editando documentos grandes](#)
- [6.2. Opciones de conversión](#)
- [7. Consideraciones finales](#)
 - [7.1. Utilidades de este tipo de edición](#)
 - [7.2. La elección del lenguaje de marcas](#)
 - [7.3. Retroalimentación](#)

1. Consideraciones previas

1.1. Introducción

Es innegable que la evolución de las nuevas tecnologías ha cambiado nuestra forma de vida, ya que ha agilizado muchos procesos tanto en nuestra vida personal como en las empresas.

Uno de los campos en los que se puede apreciar este cambio con mayor claridad es en la edición de textos. Ahora es posible editar cualquier tipo de texto sin depender del soporte físico tradicional de información (el papel). Esto significa que antes de imprimir un texto el usuario tiene la posibilidad de corregirlo y cambiar su aspecto.

En estos apuntes se hace un repaso a la forma tradicional de editar textos con un ordenador y se le enseña una alternativa fácil de usar y muy eficiente.

1.2. Conocimientos previos

Estos apuntes tratan de explicar los conceptos de una manera suficientemente clara y detallada como para que pueda aprender por usted mismo,. No obstante, se requieren unos conocimientos previos para comprender el texto con claridad, que son los siguientes:

Conocimientos mínimos del sistema operativo que maneja.

Se presupone que maneja su ordenador con soltura, de forma que sabe crear carpetas, mover y copiar archivos, manejar programas o guardar archivos en una ubicación determinada.

Conocimientos de internet.

Debe saber qué es internet, cómo se navega y cómo se descargan archivos.

Si aún no los ha adquirido se recomienda que consulte algún manual de informática básica para aprender lo que necesita. Después, siempre puede regresar a estos apuntes.

1.3. La actitud para el aprendizaje

En informática, al igual que en otros muchos aspectos de la vida, la práctica es totalmente imprescindible para poder asimilar los conceptos teóricos. Gracias a ella podrá autoevaluarse, conocer tanto lo que ya ha entendido como lo que aún no tiene tan claro y, en definitiva, mejorar continuamente.

Si lee estos apuntes del tirón y después intenta realizar lo que se explica, muy probablemente no obtenga buenos resultados y se desespere pensando que es muy complicado y que no lo podrá aprender.

Por lo tanto, lo más recomendable es que vaya practicando con los conceptos conforme van apareciendo. En concreto, configure su equipo cuando lea el capítulo de configuración. A partir de entonces, cuando el texto explique cómo se hace una determinada cosa practíquela inmediatamente. Para ello, lea cuidadosamente las instrucciones y sígalas paso por paso. Una vez que esté familiarizado con los resultados que obtiene, con la práctica logrará automatizar aquello que más utilice.

En el momento en el que lo tenga todo completamente dominado, recordará con facilidad lo que usa con más frecuencia, porque lo tendrá automatizado. Si hay algo que necesita y ha olvidado, siempre puede volver a estos apuntes para consultarlo.

2. La edición de textos

2.1. Introducción

Como se ha comentado en el apartado anterior, las nuevas tecnologías han supuesto un gran cambio en la edición de textos. Antes se dependía totalmente del soporte físico en el que tenía que estar el documento, que normalmente era el papel. La edición de textos ha supuesto dos grandes cambios.

Por un lado se ha aumentado la velocidad de escritura. Una persona podría llegar a mecanografiar a unas 300 pulsaciones por minuto, por lo que podría transcribir una conversación en el mismo momento en el que se produce.

Por otra parte, no se depende del papel para editar textos. Antes de imprimir el documento se puede hacer cualquier cambio (como corregir un error ortográfico) y se puede aplicar cualquier estilo imaginable. Esto crea una paradoja, consistente en que ahora es mucho más fácil que antes crear contenido en papel, cuando ahora prácticamente podríamos prescindir de él.

2.2. La edición de textos tradicional

En los procesadores de texto tradicionales aparece un área en la pantalla que contiene el documento que se está editando. Su principal característica es que al mismo tiempo que escribe el documento está viendo cómo va a quedar.

La ventaja que tiene esta forma de edición es que si selecciona una parte del texto y realiza un cambio de estilo (por ejemplo el tipo de letra) puede previsualizar ese cambio antes de realizarlo y lo verá al instante en el que confirme que se realice. Este es el principal argumento de quienes defienden que editar textos de esta forma es fácil.

Sin embargo, esto puede convertirse en un arma de doble filo, debido a que en su cabeza se están mezclando cuestiones de diseño y de redacción del texto. Si no le gusta algo de su documento puede parar de escribir y hacer el cambio de diseño inmediatamente, pero corre el riesgo de que se le vaya de la cabeza la idea que quería expresar.

Supongamos ahora que decide escribir todo el texto esté como esté y que aplica el diseño que le gusta cuando haya terminado de redactar. Esto hace que tenga que perder tiempo cambiando el diseño cuando la redacción ya estaba acabada.

2.3. Los lenguajes de marcas

Un lenguaje de marcas es una serie de códigos que se insertan en el texto para indicar qué parte es un título, dónde empieza y dónde termina un párrafo o cuál es cada celda de una tabla. Después esas marcas son interpretadas de forma que el resultado es un texto bien ordenado y con un buen diseño. Por lo tanto, la idea que persiguen los lenguajes de marcas es:

1. Editar el texto.
2. Interpretar las marcas.
3. Visualizar el resultado.

Esto hace que a la hora de editar el documento usted sólo tenga que preocuparse del contenido y dejar que la tecnología se ocupe del diseño.

2.3.1. Internet se escribe con un lenguaje de marcas

Internet ha revolucionado la forma de compartir información. Ahora se puede estar en contacto con cualquier persona de nuestro interés y conocer instantáneamente lo que ocurre en cualquier parte del mundo.

Sin embargo, esto es independiente de la forma de acceso de los usuarios, puesto que ahora se puede visualizar contenido de internet desde un ordenador con cualquier sistema operativo, un teléfono móvil o una tableta.

Este hecho requería que se utilizara un código con el que cualquier navegador de internet pudiera entender qué era un título, un párrafo, un enlace o una imagen con el fin de mostrarlo correctamente al usuario. Este código se llama HTML y, si ha entendido lo expuesto en el apartado anterior, comprenderá que es un lenguaje de marcas.

Este lenguaje de marcas es muy importante, por ello dedicaremos un apartado a ver cómo reconocer sus elementos cuando se muestra en la pantalla.

2.3.2. Otros lenguajes de marcas

Como verá en el capítulo dedicado a reconocer los elementos de HTML, este lenguaje es bastante complicado, ya que es poco intuitivo y ralentiza la escritura. Por ello, han surgido otros lenguajes de marcas más fáciles de aprender que luego pueden ser convertidos a HTML. En estos apuntes trataremos con dos de ellos, denominados Markdown y Txt2tags.

Conforme siga avanzando aprenderá a editar cualquier tipo de documento con estos dos lenguajes de marcas, para posteriormente convertir el trabajo a HTML y ver el resultado en el navegador o convertirlo a pdf si fuera necesario.

3. Los elementos de HTML

Este capítulo será su primer ejercicio práctico, en el que creará una página en HTML muy sencilla con el único objetivo de aprender a reconocer lo que ve en internet.

Abra un editor sencillo (como el bloc de notas en windows o gedit en linux) y escriba lo siguiente:

```
<html>
<head>
<title>Título del documento</title>
</head>
<body>
<h1>Título de nivel 1</h1>
<h2>Título de nivel 2</h2>
<h3>Título de nivel 3</h3>
<h4>Título de nivel 4</h4>
<h5>Título de nivel 5</h5>
<h6>Título de nivel 6</h6>

<p>Esto es un párrafo normal. Compárelo con los 6 niveles de título
disponibles.

<p>Ahora va una lista numerada.

<ol>
<li>Brasil.</li>
<li>Francia.</li>
<li>España.</li>
<li>Argentina.</li>
</ol>

<p>Ahora escribirá una lista sin numerar.

<ul>
<li>Una característica.</li>
<li>Otra característica.</li>
<li>Otra característica más.</li>
<li>La última característica.</li>
</ul>

<p>También hay tablas.

<table>
<tr>
```

```
<th>Título de la primera columna</th><th>Título de la segunda
columna</th>
</tr>
<tr>
<td>Dato de la primera columna</td><td>Dato de la segunda
columna</td>
</tr>
</table>

<blockquote>
Esto es una cita, que se utiliza cuando se quiere reproducir el
fragmento de un texto científico.
</blockquote>

<p>Por último, pondremos un enlace.

<a href="http://www.google.com">Busque algo en Google.</a>
</body>
</html>
```

Guárdelo como "websencilla.html", cierre el editor de textos y ábralo con el navegador de internet.

El texto que hay es suficientemente explicativo. Explore la página el tiempo que sea necesario para familiarizarse con todos sus elementos.

Es importante que lo haga porque más adelante se hará referencia a estos elementos. Si en algún momento no se acuerda de la terminología vuelva a este apartado para repasarla.

4. Las herramientas y el proceso de trabajo

4.1. Introducción

En los próximos capítulos ya se abordan los lenguajes de marcas con los que se trabajará. Para el estudio de esos capítulos se presupone que va a tener las herramientas necesarias instaladas y configuradas para poder practicar con los conceptos que se exponen.

Por lo tanto, es altamente recomendable que se asegure de tener todo el equipo listo antes de pasar al siguiente capítulo.

Esta explicación está dividida en sistemas operativos. Salte directamente a las instrucciones del que le interese.

4.2. Explicación para Windows

En Windows existe un programa que le permitirá editar los documentos y le automatizará el proceso de conversión. Este programa se llama EdSharp. En los próximos apartados sabrá instalarlo, configurarlo y utilizarlo.

4.2.1. Descarga e instalación

[Descargue EdSharp](#)

La instalación es como la de cualquier otro programa de Windows. Únicamente hay que aceptar el acuerdo de licencia y pulsar en siguiente para dejar las opciones por defecto.

En la última pantalla hay algunas opciones que puede marcar o no. Por ejemplo, existen opciones para que EdSharp abra determinados tipos de archivo, para que cree una tecla rápida o para optimizar su comportamiento con lectores de pantalla. Marque o desmarque estas casillas a su conveniencia. Si ha elegido instalar algo el programa de instalación correspondiente se abrirá automáticamente.

4.2.2. Uso de EdSharp

La interfaz del programa es como la de cualquier otro editor de textos, por lo que aquí no se enseñará su funcionamiento, pero sí que se explicarán algunas características que le ayudarán a trabajar de forma más eficiente.

4.2.2.1. Selección de trozos grandes

A continuación se explica cómo seleccionar un trozo largo de texto.

1. Sitúe el cursor en el primer carácter que va a seleccionar.
2. Pulse la tecla F8.
3. Sitúe el cursor en el final de la selección, es decir, en el primer carácter que no va a seleccionar.
4. Pulse la combinación de teclas Shift+F8.

4.2.2.2. Trabajando con listas numeradas

Si está creando una lista EdSharp tiene una forma eficiente de escribir por usted un número delante de cada elemento.

1. Escriba el primer elemento de la lista.
2. Pulse intro para pasar a la siguiente línea.
3. Repita los pasos anteriores con cada elemento de la lista.
4. Seleccione la lista.
5. Pulse el comando alt+shift+n.
6. En el cuadro de texto que aparece escriba el número desde el que se debe empezar. Por defecto está el uno.
7. Pulse el botón "ok".

4.2.2.3. Indentación

La indentación es la separación que existe entre el principio de la línea y el principio del texto.

EdSharp, entre otras cosas, está muy orientado para el trabajo con los lenguajes de programación. En algunos de ellos la indentación de ciertas líneas es imprescindible. En otros casos se usa simplemente para una mayor legibilidad.

Cuando quiera indentar una línea sitúe el cursor en cualquier parte de ella y pulse el tabulador. Si no estaba indentada EdSharp anunciará "level 1", si ya había pulsado el tabulador una vez se anunciará "level 2" y así sucesivamente. Para conocer el nivel de indentación pulse alt+i.

4.2.2.4. Creando plantillas

Si trabaja con muchos documentos en los que sólo varían unos datos pero la estructura es la misma, es posible automatizar buena parte del trabajo creando plantillas.

Una plantilla no es más que el esqueleto de un tipo de documento determinado. Cuando el usuario la invoca puede estar preparada para que EdSharp cree un formulario que pregunte por los datos que variarán. Al final de todo el proceso se genera un documento que tendrá la estructura base y los datos introducidos por el usuario en su lugar correspondiente.

Estos apuntes se centran en la creación de plantillas para documentos de texto, que son las que debe utilizar cuando trabaje con lenguajes de marcas.

La primera línea de estas plantillas es siempre la misma:

```
text form
```

Esta línea está formada por dos palabras clave y nunca aparecerá en el documento final. La palabra clave "text" indica que el resultado será un documento de texto. La palabra clave "form" indica que EdSharp debe crear un formulario que pregunte al usuario por los datos que variarán en el documento.

A partir de la segunda línea se empieza definir realmente el documento. Todo lo que debe hacer es escribir la estructura común tal y como tiene que quedar e insertar una variable por cada dato que se debe preguntar al usuario. Así, una vez que éste rellene el formulario y lo acepte EdSharp sustituirá cada variable por el valor introducido. Para crear una variable debe seguir las siguientes reglas:

- Debe escribir un signo de porcentaje (%), el nombre de la variable, un signo igual (=) y otro signo de porcentaje.
- No se admiten espacios ni caracteres especiales en el nombre de la variable.
- Tampoco debe haber ningún espacio entre el nombre y los signos.
- Entre el signo igual y el de porcentaje puede escribir el valor que tendrá la variable por defecto. En este valor sí que se admiten espacios, pero no debe haber un espacio ni entre el igual y el valor ni entre el valor y el porcentaje.

A continuación se expondrá un ejemplo muy sencillo que le enseñará cómo se crea una plantilla con sus correspondientes variables, cómo se guarda y cómo se ejecuta. Es importante que practique para reforzar los conceptos que se han expuesto y para familiarizarse con los resultados.

Supongamos que pretende planificar una serie de reuniones para el próximo miércoles y que quiere enviar un correo personalizado a cada asistente para que le comunique si el día y la hora que usted propone puede acudir a la reunión.

Lo primero que haremos es planificar cómo va a ser la estructura del texto. Supongamos que es como esta:

```
Estimado (nombre y apellidos):
```

```
Me gustaría saber si podemos quedar el próximo (día, muy probablemente miércoles) a las (hora) para hablar de (tema de la reunión).
```

```
Recibe un cordial saludo.
```

Ahora que conoce la estructura puede diseñar las variables que necesita. La siguiente tabla muestra los nombres que utilizaremos en el ejemplo y la información que se recogerá:

Nombre de la variable		Información que almacena
nombre	Nombre de la persona.	
Apellidos	Apellidos del asistente.	
dia	El día que se propone. El valor por defecto será miércoles.	
hora	La hora propuesta.	
tema	El tema a tratar.	

Ya tiene los datos suficientes como para crear la plantilla. Abra un documento en EdSharp y escriba lo siguiente:

```
text form
Estimado %nombre=% %apellidos=%:
```

```
Me gustaría saber si podemos quedar el próximo %dia=miércoles% a las %hora=% para hablar de %tema=%.
```

```
Recibe un cordial saludo.
```

Para guardar esta plantilla pulse alt+s. En el cuadro de texto que aparece está escrita la ruta en la que se guardan las plantillas y el nombre que EdSharp sugiere que es

"NoName1.txt". Pulse la tecla fin, borre con la tecla de retroceso "NoName1.txt", escriba reuniones.txt y pulse intro. Ya puede cerrar EdSharp.

A continuación se aplicará la plantilla que acaba de crear. Para ello abra un documento nuevo y pulse el comando alt+v. Aparecerá una lista con todas las plantillas disponibles. Seleccione "reuniones.txt" y pulse intro.

En el formulario que aparece verá que todos los cuadros de texto están vacíos excepto el del día, que por defecto tiene escrito miércoles. Rellene el formulario con los siguientes datos:

nombre
 Alberto
apellidos
 Zaragoza Muñoz
día
 miércoles
hora
 12:30
tema
 la compra de la nueva máquina.

Una vez que ha rellenado los datos pulse el botón "OK" y observe cómo queda el texto. Si quisiera enviárselo a Alberto sólo tendría que pulsar CTRL+m. En su programa de correo electrónico debería escribir únicamente la dirección y el asunto que le interesan, debido a que en el cuerpo ya aparece el texto del documento.

4.2.3. Configurando EdSharp

Para acceder a la configuración de EdSharp Haga click en "Misc" en la barra de menú y después en "manual options", o pulse alt+shift+m. En el diálogo que se le abre pulse el botón "main". Se le abrirá el archivo de configuración.

Este archivo está dividido en secciones. Cada una de ellas comienza con su título entre corchetes. Dentro de ellas, cada aspecto que se puede configurar se estructura como una clave que hace referencia a lo que se quiere configurar, un signo igual y un valor. De tal manera que la estructura del archivo es como esta:

```
[sección]
configuracion=valor
otracosa=otrovalor
[otrarsección]
unaspecto=unvalor
```

Lo primero que va a cambiar es una línea que está en la sección "[Options]". La línea original dice:

```
YieldEncoding=""
```

La línea con la modificación realizada es:

```
YieldEncoding="UTF-8"
```

Esta línea es importante para que después de la conversión se visualicen correctamente los caracteres acentuados y las ñes.

En la sección "[import]" están las distintas opciones de conversión entre formatos. Exploremos una de ellas:

```
md2html="%ProgDir%\Convert\Pandoc\bin\pandoc.exe -f markdown -t html  
-s %Source% -o %Target%"
```

Esta línea, con las modificaciones que le interese realizar es la que hará que EdSharp convierta sus documentos hechos con Markdown a HTML.

La primera parte (la que está a la izquierda del signo igual) dice que se convertirá de markdown (md) a html. Para ello deberá guardar sus documentos con la extensión "md".

Lo que hace el resto de la línea es llamar al programa pandoc que está instalado en la misma carpeta de EdSharp. Si observa se escribe "%ProgDir%" que es la carpeta en la que está EdSharp y a partir de ahí se escribe toda la ruta hasta el programa.

Además debe conocer que la variable "%Source%" hace referencia al archivo que acaba de editar y la variable "%Target%" Sirve para que EdSharp abra el archivo resultado de la conversión.

El resto de la línea son opciones propias del programa que ya irá apendiendo conforme estudie estos lenguajes.

Cuando acabe de hacer las modificaciones en el fichero de configuración guarde los cambios y cierre EdSharp.

4.2.4. El proceso de conversión

A continuación va a aprender cómo se edita un documento simple y cómo se convierte a HTML. Para este ejemplo usaremos Markdown, pero para Txt2tags debe seguir el mismo proceso.

Lo primero que va a tener que hacer es abrir EdSharp y editar el documento cumpliendo las reglas del lenguaje de marcas que vaya a utilizar. En este caso, escriba este texto de ejemplo:

```
% Prueba
```

```
Este es mi primer documento de prueba utilizando Markdown.
```

Note que entre las dos líneas de texto hay una línea en blanco.

Una vez acabado el documento tiene que guardarlo con la extensión de ese lenguaje de marcas. En este caso, guárdelo como "prueba.md" en la ubicación que elija. Cuando lo haya hecho cierre EdSharp.

A continuación, haga doble click o pulse intro sobre el documento que acaba de guardar. Si es la primera vez que lo hace, se le preguntará con qué programa desea abrirlo. Busque el programa EdSharp y asócielo.

Una vez que se abra EdSharp se le abrirá un cuadro de diálogo preguntándole en qué formato desea el archivo. Aquí trabajará con dos opciones:

- Elija html si quiere convertirlo.
- Elija md (la extensión con la que lo había guardado) si quiere realizar un cambio.

En este caso, como lo que se pretende es convertir el archivo elija html. Se le abrirá el archivo html en EdSharp. Simplemente guárdelo como "prueba.html".

Por último, tras cerrar EdSharp abra "prueba.html" y visualice el resultado en su navegador de internet habitual.

Por tanto, para que le quede más claro, a continuación se hace un resumen del proceso de edición y conversión con EdSharp cuando se trabaja con lenguajes de marcas:

1. Abra un documento nuevo en EdSharp y escriba el texto.
2. Guarde el documento de acuerdo a las siguientes reglas:
 - Si usa Markdown la extensión es "md".
 - Si usa Txt2tags la extensión es "tt".
3. Cierre Edsharp.
4. Localice el fichero guardado y haga doble click sobre él.
5. En el diálogo que aparece haga una de estas dos cosas:
 - Elija la extensión con la que lo había guardado si quiere hacer algún cambio.
 - Elija html si quiere convertirlo.
6. Una vez convertido guarde el resultado con la extensión "html".
7. Ya puede abrirlo con su navegador de internet.

4.3. Explicación para otros sistemas operativos

En otros sistemas operativos no existe una herramienta que lo haga todo de forma automatizada como en Windows. Si quiere utilizar estos lenguajes de marcas tendrá que conseguir un editor que sea capaz de guardar en formato UTF-8 o una herramienta que pueda convertir a esta codificación. Después deberá tener instalado el conversor necesario y utilizar la línea de comandos (también llamada consola) para realizar la conversión. A continuación se proporcionan unos enlaces que le darán más información:

- [Conversor de Markdown](#)
- [Proyecto Txt2tags](#)

5. Markdown

Ahora que ya ha configurado todo el equipo y sabe cómo convertir sus documentos es hora de empezar a hablar de las reglas propias de cada lenguaje de marcas.

En este capítulo se hablará sobre Markdown y en el siguiente sobre Txt2tags. Primero se hablará de la sintaxis, es decir, qué símbolos hay que insertar en el texto dependiendo de lo que vaya a hacer y después se hará referencia a las opciones de conversión que puede cambiar en la configuración de EdSharp.

5.1. La sintaxis

5.1.1. La estructura del documento

La estructura del documento es muy sencilla. Se escribe un encabezado, se deja una línea en blanco y se escribe el contenido del documento, también llamado cuerpo.

5.1.1.1. El encabezado

El encabezado son tres líneas que se inician con el signo de porcentaje (%), después un espacio y después el texto que corresponda de acuerdo a las siguientes reglas:

- La primera línea debe contener el título del documento.
- La segunda línea contendrá el nombre del autor.
- La última línea corresponde a la fecha de creación del documento.

El encabezado nos da la posibilidad de no incluir todos los datos. Para ello haga lo siguiente:

- Si no queremos incluir ni el autor ni la fecha basta con escribir únicamente la línea del título.
- Si lo que pretende es incluir el título y el autor, pero no quiere que aparezca la fecha escriba únicamente las líneas dedicadas al título y el autor.
- Si lo que quiere es incluir el título y la fecha sin hacer referencia al autor escriba la línea del título, escriba únicamente el signo de porcentaje en la segunda línea y luego escriba la fecha en la última línea.

5.1.1.2. Ejemplos prácticos

A continuación se expone una serie de ejemplos que le ayudarán a diferenciar el encabezado del resto del documento y le mostrarán los distintos tipos de encabezado.

El primer ejemplo muestra un encabezado completo y el cuerpo del documento.

```
% Experimentación
% Pedro García
% 15/09/2007
```

Este documento demuestra que ha finalizado el experimento que pretendía contrastar la hipótesis que inicialmente se había planteado.

Observe las tres líneas que conforman el encabezado, la línea en blanco de separación y el cuerpo del documento.

A continuación se expone un documento que no tiene autor ni fecha.

```
% Cosas prohibidas
```

```
Este documento contiene información obtenida de forma ilegal y acusaciones sin pruebas. Por eso no lleva la firma del autor.
```

A continuación se muestra un ejemplo de un documento al que le falta la fecha.

```
% Título  
% autor
```

```
Y aquí iría el cuerpo del documento.
```

Finalmente, se muestra un documento a cuyo encabezado le falta el nombre del autor.

```
% Manual de instrucciones  
%  
% 20/11/2005
```

```
Escriba aquí las instrucciones.
```

Con esto ya conoce todas las posibilidades que le ofrece el encabezado. Los subapartados que siguen corresponden al cuerpo.

5.1.2. Capítulos, apartados y subapartados

Para una correcta legibilidad de cualquier documento, éste necesita ser jerarquizado en capítulos, apartados y subapartados. Para llevar a cabo esta tarea existe lo que en el apartado de html hemos denominado títulos de nivel.

Como se vio en ese capítulo existen seis títulos de nivel disponibles. Para crear un nuevo título haga lo siguiente:

1. Deje una línea en blanco entre el texto anterior y el título.
2. Escriba el signo de almohadilla (#) tantas veces como el nivel que quiera para el título. Recuerde que el nivel 1 es el mayor en importancia.
3. Deje un espacio y escriba el texto para ese título.
4. Deje una línea en blanco y escriba el texto de ese apartado.

En el ejemplo que figura a continuación, encontrará el final de un apartado y el inicio del siguiente, cuyo título es de nivel 3.

Y aquí finaliza el apartado.

Nivel 3

Este es el texto que pertenece al siguiente apartado.

5.1.3. Párrafos y saltos de línea

Un párrafo se separa de otro por una línea en blanco. Si en algún momento quiere forzar un salto de línea debe escribir dos espacios y un salto de línea (pulsando la tecla intro) donde quiera que se produzca. A continuación tiene un ejemplo.

```
Este párrafo tiene muchos saltos de línea
y sería ideal para la poesía,
pero yo no tengo futuro en esto
porque los versos ni riman ni tienen métrica.
```

Este otro párrafo no tiene saltos de línea forzados, porque escribo en prosa, que es como se escribe normalmente.

5.1.4. Las listas

Según lo que debería conocer de HTML, existen dos tipos de listas, las numeradas y las no numeradas.

Si quiere escribir una lista no numerada, escriba un guión (-), un signo más (+) o un asterisco. Luego escriba un espacio y luego el texto. Escriba cada elemento en una línea y finalice la lista dejando una línea en blanco entre su último elemento y el texto siguiente.

Es indistinto utilizar el guión, el más o el asterisco para las listas. Sin embargo, si la distribución de su teclado está en español le será más fácil utilizar el guión por la velocidad de escritura.

Una lista de colores se escribiría así:

- Rojo
- Verde
- Amarillo
- Azul

Si lo que quiere es crear una lista numerada, lo que debe anteponer a cada elemento es un número seguido de un punto. La lista anterior numerada quedaría así:

1. Rojo
2. Verde
3. Amarillo
4. Azul

Aquí puede utilizar la enumeración de elementos que tiene EdSharp, que se explicó en el apartado dedicado a este programa.

5.1.4.1. Listas anidadas

Puede anidar una lista dentro de otra. Para ello, sólo tiene que escribir el elemento en el que irá la lista anidada, escribir cada elemento de la lista anidada con un nivel superior de indentación y seguir con los elementos de la lista original cuando termine. Aquí tiene un ejemplo:

- Colores
 - Rojo
 - Verde
 - Amarillo
 - Azul
- Formas
- Tamaños

Recuerde que EdSharp también le ayuda con la indentación.

5.1.5. Enlaces

Para crear un enlace abra un corchete, escriba el texto en el que se hará click para activar el enlace, cierre el corchete, abra un paréntesis, escriba la dirección y cierre el paréntesis.

Si recuerda "websencilla.html" sabrá que contenía un enlace al buscador Google. Veamos cómo se crearía este enlace usando Markdown.

```
[Busque algo con Google] (http://www.google.com)
```

5.1.6. Imágenes

Si ha comprendido cómo se crean los enlaces, el proceso para crear una imagen es muy sencillo. Debe escribir un signo de cerrar admiración (!), a continuación la descripción de la imagen entre corchetes y, finalmente, la ruta y el nombre completo del archivo de imagen entre paréntesis.

5.1.7. Citas

Como se vio en el capítulo de HTML las citas se utilizan para copiar fragmentos de textos científicos.

Para crear una cita anteponga un signo de mayor que (>) y un espacio al texto. Por ejemplo:

```
> Esto es una cita que dijo alguien famoso.  
>  
> Como hablaba mucho, el texto tiene varios párrafos. Por eso una de  
las líneas sólo tiene el mayor que.
```

EdSharp tiene una forma de ayudarle a escribir las citas. Para ello, seleccione el texto que debe formar parte de la cita y pulse CTRL+q.

5.1.8. Enfatizando textos

Markdown permite escribir cierta parte del texto en negrita o en cursiva. Para ello hay que encerrar el texto entre grupos de asteriscos. Un único asterisco significa negrita, 2 significa cursiva y 3 significa negrita y cursiva. A continuación se expone un ejemplo:

```
*Esto va en negrita.*  
**Esto va en cursiva.**  
***Esto va en negrita y cursiva.***
```

Observe que no se ha dejado ningún espacio entre el texto y los asteriscos.

5.1.9. Código sin procesar

En algunas ocasiones puede que le interese tener un trozo de texto que debe quedar tal y como lo escriba, es decir, sin que se interprete ningún tipo de marca. Para ello debe usar el caracter tilde (~) que se obtiene pulsando alt gr y el número 4. La forma de hacerlo es la siguiente:

1. Escriba una línea formada por, al menos, tres tildes.
2. A partir de la línea siguiente escriba el texto cuyas marcas no deban interpretarse.
3. Finalmente, en la última línea escriba tantas tildes como puso en la primera línea.

5.1.10. Creando el índice

El índice se crea utilizando un tipo de enlace que se denomina "enlace a la misma página". Este tipo de enlace sirve, como su propio nombre indica, para saltar a una parte concreta de una página. Así, el índice no es más que un conjunto de enlaces que irán saltando a los distintos apartados del documento. Para ello, cree un enlace normal con las siguientes reglas:

- Como texto del enlace se sugiere que escriba el título del apartado.
- Lo que debe escribir entre paréntesis es un signo de almohadilla seguido por el título del apartado sustituyendo los espacios por guiones, pero respetando mayúsculas y caracteres acentuados.

Así, si el título del apartado es:

```
## La historia que cambió una vida
```

El enlace en el índice sería:

```
[La historia que cambió una vida](#La-historia-que-cambió-una-vida)
```

5.1.11. Caracteres especiales

Suponga que quiere que un caracter de la sintaxis de Markdown no sea interpretado como tal, sino que quiere que ese caracter se escriba en el texto. Basta con anteponer una barra invertida (\) al caracter en cuestión.

Por ejemplo, esto sería útil cuando quiere escribir un asterisco y no desea iniciar un texto en negrita o cuando quiere escribir un número seguido de un punto y no quiere que accidentalmente se cree una lista.

Tenga en cuenta que si quiere escribir la barra invertida deberá escribirla dos veces. Por ejemplo una ruta a una carpeta se escribiría así:

```
c:\\una carpeta\\otra carpeta
```

5.1.12. Otros elementos

Si quiere escribir otros elementos no soportados por este lenguaje de marcas tendría que usar directamente HTML. A continuación se deja un enlace a un tutorial sobre este lenguaje.

[Tutorial de HTML](#)

5.2. Opciones de conversión

La tabla siguiente muestra algunas opciones del conversor de Markdown. Compruebe las que tenga en la configuración de EdSharp (si lo utiliza en Windows) y cambie las que le interese. Si usa otro sistema operativo acuda a los recursos que se han proporcionado en la sección correspondiente.

Opción		Significado
-f	Es el formato de origen. En este caso, Markdown.	
-t	Es el formato de destino. En nuestro caso HTML.	
-o	Hace referencia a cómo se llamará el fichero de salida. En la configuración de EdSharp debe poner %Target%	
-s	Hace que se muestre el encabezado del documento. Si no estuviera, sólo se mostraría el cuerpo.	
-S	Sirve para corregir la puntuación. Pruebe las diferencias y úsela según su conveniencia.	

6. Txt2tags

Txt2tags es el otro lenguaje de marcas que se trata en estos apuntes. Su filosofía es parecida a Markdown, aunque las marcas no son las mismas.

Otra diferencia está en la forma de guardar los archivos. Si antes usaba la extensión "md" para Markdown ahora usará la extensión "tt" para Txt2tags.

6.1. La sintaxis

6.1.1. La estructura del documento

Al igual que en Markdown puede escribir un encabezado que contenga el título, el autor y la fecha. Luego se deja una línea en blanco y se empieza con el cuerpo del documento.

Aquí tiene unas líneas de ejemplo:

```
Título
Autor
11/05/2003
```

```
Este es el contenido del documento.
```

Observe que, a diferencia de Markdown, no se ha puesto el signo de porcentaje al principio de cada línea, sino que estas líneas se consideran de encabezado por ser las primeras del documento.

6.1.1.1. Particularidades de la fecha

Txt2tags le permite insertar automáticamente la fecha en la que se hizo la última modificación en el formato que elija. Para ello escriba `%%mtime` y entre paréntesis escriba "%d" donde vaya el día, "%m" donde deba ir el mes y "%y" donde quiera que vaya el año escrito con dos cifras o "%Y" si prefiere que el año se escriba con cuatro dígitos. Además, puede escribir %A para que aparezca el día de la semana (en inglés) y %B para que aparezca el nombre del mes (también en inglés).

A continuación se muestra un ejemplo de encabezado que contiene la fecha de última modificación en el formato día/mes/año.

```
Procedimiento de trabajo
Jefe de departamento
Última modificación: %%mtime(%d/%m/%Y)
```

6.1.1.2. El área de configuración

De forma opcional puede insertar líneas de configuración entre el encabezado y el cuerpo.

Para cada aspecto que quiera configurar deberá escribir el signo de porcentaje, un signo de cerrar admiración, un nombre preestablecido que se refiere a lo que va a configurar, un signo de dos puntos, un espacio en blanco y el valor que tendrá. Esto le quedará claro con los distintos ejemplos que se exponen a continuación.

La primera orden de configuración de la que hablaremos será "Style". Usted puede guardar un [archivo CSS](#) en la misma carpeta para luego especificar que su documento tendrá ese estilo. Para ello, si el archivo se llama migranestilo.css en el área de configuración escribirá:

```
%!Style: migranestilo.css
```

Para finalizar, hablaremos de las órdenes PreProc y PostProc. Estas dos órdenes se dedican a buscar un texto y reemplazarlo por otro. La diferencia es que PreProc actúa antes de la conversión y PostProc actúa después.

Si se reemplazará una palabra por otra se pueden separar simplemente con un espacio en blanco como en el siguiente ejemplo:

```
%!PreProc: naipe carta
```

En este ejemplo, antes de que el archivo sea convertido se sustituirá la palabra naipe por la palabra carta. Tenga en cuenta que todas las sustituciones que se hagan con estas dos reglas afectan a todo el documento, por lo que no es posible decidir en qué parte se realiza o no la sustitución.

Si, por el contrario, una de las dos expresiones (la que se va a buscar y la que la reemplaza) tienen más de una palabra se deberán encerrar entre comillas simples (''). Como ejemplo, supongamos que queremos declarar que el documento HTML está en español. La regla de configuración sería la siguiente:

```
%!PostProc: '<HTML>' '<HTML lang="es">'
```

El área de configuración debe ir justo después del encabezado y finaliza en la línea en blanco que la separa del cuerpo.

Para que le quede más claro vamos a ver un ejemplo práctico. Supongamos que hemos creado una hoja de estilo CSS. Además, supongamos que hemos especificado la fecha para que nos diga el nombre del día (con %A) y el nombre del mes con (%B), pero como sabemos que esto se escribirá en inglés y nuestro documento está en castellano nos gustaría que se hiciera la traducción. Este documento contiene el encabezado, la configuración y el inicio del cuerpo para que aprecie con claridad dónde aparece cada área para que se interprete de forma correcta.

```
Documento importante
Autor excepcional
Modificado por última vez el %%mtime(%A, %d de %B de %Y)
%!Style: migranestilo.css
%!PostProc: lunes lunes
%!PostProc: martes martes
%!PostProc: miércoles miércoles
%!PostProc: jueves jueves
%!PostProc: viernes viernes
%!PostProc: sábado sábado
```

```
%!PostProc: domingo domingo
%!PostProc: enero enero
%!PostProc: febrero febrero
%!PostProc: marzo marzo
%!PostProc: abril abril
%!PostProc: mayo mayo
%!PostProc: junio junio
%!PostProc: julio julio
%!PostProc: agosto agosto
%!PostProc: septiembre septiembre
%!PostProc: octubre octubre
%!PostProc: noviembre noviembre
%!PostProc: diciembre diciembre
```

Y aquí empieza el cuerpo del documento.

Con esto ya tiene todo lo que debe saber sobre las áreas de encabezado y configuración. El resto del capítulo se centra en el cuerpo del documento.

6.1.2. Capítulos, apartados y subapartados

En Txt2tags existen los seis niveles de título disponibles que ya conoce. Existen dos diferencias con respecto a Markdown.

Por un lado deberá escribir una serie de signos al principio del título y la misma serie al final.

La segunda gran diferencia es que podrá controlar si quiere que el título sea numerado automáticamente (como en estos apuntes) o no.

Todo lo que tiene que hacer es escribir el signo al principio y al final del título tantas veces como nivel quiere que tenga el título. Así, si el título será de nivel 3 escribirá tres símbolos, un espacio, el texto del título, otro espacio y los mismos tres símbolos que al principio.

Use el signo de igual (=) si no quiere que se numeren los títulos. Use el signo más (+) si quiere que se numeren los títulos.

Si decide numerar los títulos tenga en cuenta las siguientes reglas:

- Si quiere crear el apartado 1 use el título de nivel 1.
- Si quiere crear el subapartado 1.1 use un título de nivel 2.
- Si quiere crear el subapartado 1.1.1 use el nivel 3.
- Cuando quiera crear el apartado 2 vuelva a usar el nivel 1.
- Si, por ejemplo, usa un nivel 2 antes de el primer título de nivel 1 se crearía el apartado 0.1.

Por último, deje dos líneas en blanco antes y después del título.

A continuación van unos ejemplos de títulos numerados y sin numerar.

```
==== Título de nivel 4 sin numerar ====
```

+ Título de nivel 1 numerado +

6.1.3. Párrafos

Al igual que en Markdown, los párrafos se separan unos de otros con una línea en blanco. Tenga en cuenta que los párrafos también separan las marcas (por ejemplo el título del texto o el final de una lista del resto del documento). Si no respeta estas reglas, ningún lenguaje de marcas interpretará los símbolos como pretendía.

Los saltos de línea no tienen una marca establecida, pero puede inventar una usando directivas de configuración y conociendo el código HTML.

El código HTML para el salto de línea es `
`. Ahora sólo le queda inventarse la marca. Debe tener la precaución de elegir un signo que no esté entre las marcas de Txt2tags. Supongamos que quiere que la marca sean dos arrobas consecutivas. Todo lo que tendría que hacer sería incluir esta regla en el área de configuración:

```
%!PostProc: @@ '<br>'
```

En este momento ya puede escribir `@@` en cualquier parte de su documento y se generará un salto de línea.

6.1.4. Listas

Para crear una lista no numerada anteponga un guión y un espacio a cada elemento de la lista.

Si lo que quiere es crear listas numeradas debe anteponer un signo más. Esto le ahorra tener que escribir cada uno de los números.

Para anidar listas inserte un espacio antes del guión o del más de cada elemento de la lista anidada. No lo confunda con la forma de trabajar de Markdown en la que se indentan las listas anidadas pulsando el tabulador.

Txt2tags también permite crear otro tipo de listas llamado listas de definición. Cada elemento de la lista está formado por una pareja de subelementos. El primero corresponde al término y el segundo a la definición. Para crear cada elemento de la lista siga estas reglas:

- En la primera línea escriba el signo de dos puntos, un espacio en blanco y el término.
- En la segunda línea escriba dos espacios y la definición.

A continuación se muestra un ejemplo de creación de este tipo de listas. Fíjese en cómo están encarados entre sí los elementos de la lista (signos de dos puntos) y cómo se encaran entre sí cada término con su definición.

- : Cubo.
Figura geométrica formada por seis caras.
- : Pirámide.
Figura geométrica que tiene una cúspide.
- : Cilindro.
Figura geométrica que no tiene vértices.

Deje dos líneas en blanco cuando finalice la lista.

6.1.5. Enlaces

Escribir un enlace en Txt2tags es muy sencillo. Sólo tiene que escribir entre corchetes el texto del enlace, un espacio y la dirección. Por ejemplo:

```
[Busque algo con Google http://www.google.com]
```

6.1.6. Imágenes

Escriba entre corchetes el nombre del archivo que contenga la imagen. No se permiten espacios. Actualmente no existe ninguna forma sencilla de proporcionar una descripción alternativa de las imágenes que mejore la accesibilidad. Si quiere proporcionar esta descripción debería realizar una primera conversión para conocer el código HTML de la imagen. Después debería insertar una línea por cada imagen en el área de configuración. En esta línea debe usar PostProc para sustituir el código original por el código que incluya la descripción de la imagen.

6.1.7. Destacando textos

Al igual que en Markdown el texto se encierra entre grupos de marcas. No debe haber espacios ni saltos de línea entre las marcas y el texto. No obstante, las marcas se pueden combinar encerrando unas dentro de otras. Los símbolos disponibles son:

- Dos asteriscos para la negrita.
- Dos barras para la cursiva.
- Dos barras bajas () para el subrayado.

6.1.8. Citas

Para crear una cita indente el texto que quiere que forme parte de la misma. Tenga en cuenta que no es posible incluir múltiples párrafos dentro de una cita.

6.1.9. Tablas

La forma de crear tablas consiste en definir cada fila en una línea distinta y encerrar el texto entre delimitadores de celda. Los delimitadores de celda son la doble barra vertical (||) para los títulos de las columnas y la barra vertical (|) para el resto de las celdas.

Como ejemplo vamos a crear una tabla de dos columnas con algunos nombres de países y sus correspondientes capitales.

```
|| País || Capital ||
| Inglaterra | Londres |
| Francia | París |
| Italia | Roma |
```

6.1.10. Líneas horizontales

Son líneas que sirven de separación, por ejemplo, entre el final de un apartado y el principio del siguiente.

Para crear una línea de este tipo, escriba en una línea en blanco, al menos 20 guiones (-) si quiere que la línea sea fina y al menos 20 signos de igual (=) si quiere una línea gruesa.

6.1.11. Código sin procesar

Es ese código en el que no quiere que se interpreten las marcas o que simplemente quiere dejar tal cual escribe, por ejemplo, forzando saltos de línea.

El delimitador de este código en Txt2tags es el acento grave (`), pero se usa de formas distintas dependiendo de lo que quiera dejar sin procesar:

- Si lo que pretende es dejar sin procesar una única palabra deberá encerrarla con dos acentos graves a la izquierda y dos a la derecha sin dejar ningún espacio en blanco entre la palabra y estos acentos.
- Si lo que quiere dejar sin procesar es una línea escriba tres acentos graves, un espacio en blanco y el contenido de la línea.
- Si lo que no se procesará es un bloque formado por dos o más líneas, debe escribir antes y después de este bloque una línea formada por tres acentos graves.

6.1.12. Creando el índice

Escriba "%toc" donde quiera que aparezca el índice, teniendo en cuenta las siguientes reglas:

- Sólo debe aparecer eso en esa línea.
- Todo título, independientemente de si está numerado o no aparecerá en el índice.
- Cuando realice la conversión debe usar la opción "--toc" para que funcione.

Por ejemplo, si quiere hacer aparecer el índice justo después del encabezado, siga este ejemplo:

```
Texto con índice
Autor brillante
Actualizado el %%mtime(%d/%m/%Y)
```

```
%%toc
```

Aquí va el texto del documento.

Note que hay dos líneas en blanco entre la última línea del encabezado y "% %toc".

6.1.13. Editando documentos grandes

Si va practicando con este lenguaje como ya se ha ido recomendando a lo largo de los apuntes, se dará cuenta de lo fácil que es escribir cuando se recuerdan las marcas. Sin embargo, puede ser complicado llevar a cabo una actualización, debido a que cuanto más grande sea el documento más tiempo tardará en encontrar el lugar en el que debe realizar la modificación.

La forma que tiene Txt2tags de solucionar esto es permitirle trabajar con varios archivos por separado e insertar el contenido de un archivo en otro donde le interese con un comando muy sencillo. Este comando se llama `%!Include` (incluir en inglés) y su sintaxis es la siguiente:

```
%!Include: nombreadarchivo.tt
```

El nombre del archivo puede contener espacios, pero no caracteres especiales (como letras acentuadas).

Para que al ejecutarse esta orden se interpreten todas las marcas de forma adecuada debe tener en cuenta algunas consideraciones. Lo que hace esta orden es interpretar el contenido de acuerdo a las reglas de Txt2tags e incluirlo en el lugar en el que aparezca esta orden. Esto significa que las tres primeras líneas del archivo a incluir se considerarán encabezado y no se incluirán. Para evitar esto, si el archivo no empieza con el título de algún apartado deje una línea en blanco al principio del archivo. Si por el contrario el archivo empieza con un título debe dejar tres líneas en blanco al principio (la del encabezado y las dos que deben ir antes del título).

Por otra parte, si un archivo finaliza con una lista y va a incluir después el contenido de ese archivo debe dejar tres líneas en blanco al final del primer archivo para que Txt2tags cierre la lista.

Como ejemplo práctico, vamos a suponer que está escribiendo un libro que contiene varios capítulos.

Para trabajar cómodamente, es recomendable que cree una carpeta en la que irá guardando los distintos archivos que conformarán el libro.

A partir de ahí, lo que puede hacer es crear un archivo por cada capítulo del libro. Como ejemplo, vamos a incluir aquí el inicio del primer capítulo.

```
= Capítulo 1. El principio =
```

Este texto es el inicio de una gran historia. Está extraordinariamente bien escrito, de tal forma que engancha al lector desde el primer momento.

Observe que hay tres líneas en blanco al principio. La primera de ellas indica a Txt2tags que el encabezado de este archivo está vacío. Las otras dos líneas están ahí porque la orden "%!Include" simplemente escribe el contenido del archivo a incluir como si usted lo hubiera copiado y pegado manualmente. Por lo tanto, las dos primeras líneas que se insertarán de ese capítulo son las dos líneas en blanco que separan el título del contenido anterior.

Supongamos que el libro tiene 9 capítulos y que cada uno de ellos está guardado en un archivo con los nombres "cap1.tt", "cap2.tt", "cap3.tt", "cap4.tt", "cap5.tt", "cap6.tt", "cap7.tt", "cap8.tt" y "cap9.tt". Sólo le quedaría escribir el archivo principal del libro. Supongamos que contendrá el título, el autor, la fecha de modificación, el índice y los capítulos del libro. El código sería el siguiente:

```
Titulo del libro
Autor
%%mtime (%d/%m/%Y)
```

```
%%toc
```

```
%!Include: cap1.tt
%!Include: cap2.tt
%!Include: cap3.tt
%!Include: cap4.tt
%!Include: cap5.tt
%!Include: cap6.tt
%!Include: cap7.tt
%!Include: cap8.tt
%!Include: cap9.tt
```

Como ya se ha comentado, la principal ventaja es que cualquier cambio en un capítulo le requerirá menos tiempo, porque en el archivo específico sólo está ese capítulo y la búsqueda será más ágil que si tuviera que encontrar algo entre todo el libro.

Otra gran ventaja es que puede probar la conversión de un capítulo por separado. Por ejemplo, si en alguno de ellos ha insertado una nueva marca y no tiene claro cómo le quedará, puede convertir únicamente ese capítulo y ver los resultados.

Por último, esta forma facilita mucho el trabajo entre varios autores. Suponga que algún capítulo debe escribirlo alguien que usted conoce que es especialista en la materia. Podría enseñarle lo principal sobre este lenguaje de marcas y dejarle que se concentre únicamente en su capítulo desde el diseño y redacción hasta las posibles actualizaciones.

6.2. Opciones de conversión

Esta tabla muestra las opciones de conversión que puede utilizar:

Opción		Significado
-t	Especifica el formato de destino. en este caso debe ser HTML.	
-o	Especifica el nombre de archivo de destino. En la configuración de EdSharp debe poner "%Target%".	
-- toc	Crea el índice del documento.	

7. Consideraciones finales

7.1. Utilidades de este tipo de edición

La primera utilidad clara de este tipo de edición es abrir el resultado final con cualquier dispositivo. Recuerde que los archivos html se abren con cualquier navegador de internet y que cualquier sistema tiene uno por defecto.

La otra utilidad es abrirlo con un procesador de textos (como Word 2007 o bien OpenOffice) y convertir el archivo a pdf. En las opciones de configuración es imprescindible que deje marcada la casilla que haga referencia a incluir etiquetas para una mayor accesibilidad.

7.2. La elección del lenguaje de marcas

Ambos lenguajes de marcas son perfectamente utilizables. Este texto no pretende demostrar si uno es mejor que el otro. La idea es que los aprenda y se decida según lo que necesite hacer, lo que le parezca más sencillo o lo que haya aprendido mejor.

Recuerde que la mejor manera de aprender lo que se explica en estos apuntes es practicando los conceptos conforme los va leyendo. Por lo tanto, cualquier documento que necesite editar (un currículum, un manual, una novela...) pruebe a redactarlo con uno de estos lenguajes de marcas. Así irá ganando soltura y, aunque se pueda atascar al principio, poco a poco irá mejorando.

No se preocupe por no recordar todas las marcas. Seguramente sabrá con facilidad las que más utilice. Siempre tiene estos apuntes para repasar lo que ha olvidado.

7.3. Retroalimentación

Desde el principio de estos apuntes se ha dicho que el propósito era explicar los conceptos de la manera más clara posible. Sin embargo, el autor nunca puede conocer esto si los usuarios finales no se lo comunican. Por lo tanto, se espera cualquier duda o sugerencia para mejorar al correo electrónico quicosv (arroba) gmail (punto) com (escrito así para evitar spam).