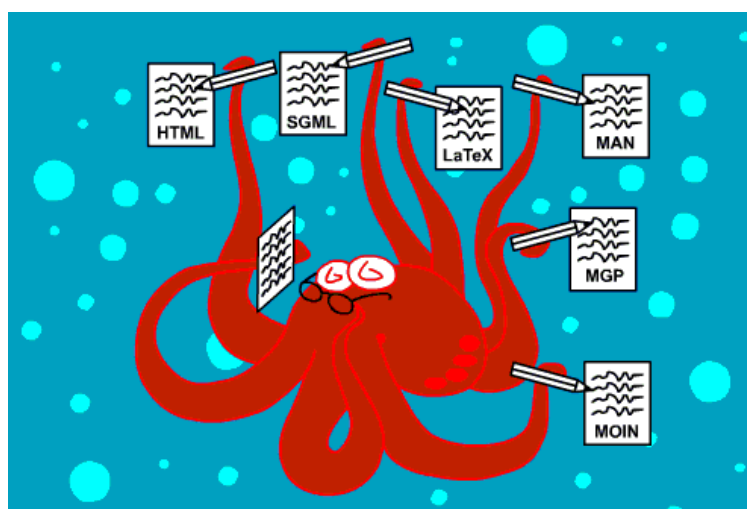


Guia del Usuario de Txt2tags

<http://txt2tags.sf.net>



Aurélio Marinho Jargas v2.3 (May 2005)

Traducción al español: Antoni Serra Devecchi (Júl 2005)

Contenido

Parte I – Introducción a Txt2tags.....	1
Tus Primeras Preguntas.....	1
Estructuras de Formato Soportadas por txt2tags.....	2
Formatos de salida soportados.....	3
Estructuras Soportadas en los Archivos finales.....	4
Las tres interfaces de usuario: GUI, Web y Línea de Comandos.....	5
Parte II: – OK!, Quiero utilizar el programa. ¿Que debo hacer ?.....	9
Obtener e instalar Python.....	9
Obtener Txt2tags.....	9
Instalar txt2tags.....	9
Instalación de los Archivos de Sintaxis para los Editores de Texto.....	10
Parte III – Escribiendo y convirtiendo vuestro primer documento.....	11
Probar las herramientas.....	11
Escribir el encabezamiento del documento.....	11
La Primera Conversión : La Interfaz Gráfica.....	11
La Primera Conversión: Interface en Línea de Comandos.....	12
Comprobar el resultado.....	13
Escribir el cuerpo del documento.....	13
Parte IV – Dominando los Conceptos de Txt2tags.....	15
Las Áreas de un Documento .t2t.....	15
Área de Encabezamiento.....	16
Area de Configuración.....	16
Área del Cuerpo del Documento.....	17
Configuraciones: ¿Cómo Configurar algo?. ¿Cuál es la sintaxis?.....	17
Opciones en Línea de Comando.....	18
Archivo de Configuración del Usuario (RC).....	18
Orden de Aplicación de las Configuraciones y Precedencia.....	19
El Comando %!include.....	19
El Comando %!includeconf.....	21
Part V – Dominando las Marcas.....	23
Encabezamiento.....	23
Título y Título Numerado.....	24
Párrafo.....	24
Comentario.....	24
Modificadores de Fuentes: Negrita, Itálica y Subrayado.....	24
Fuente no Proporcional (Monoespaciada).....	25
Línea Verbatim, Zona Verbatim.....	25
Línea de Separación , Línea Gruesa.....	26
Enlaces, Enlaces nombrados.....	26
Cita.....	26
Listas, Listas Numeradas, Listas de Definición.....	27
Imágen.....	27
Tabla.....	27
Raw,Líneas y Zonas Raw (tal-cual).....	28
Parte VI – Dominando las Macros.....	29
%%date.....	29
%%mtime.....	30
%%infile.....	30
%%outfile.....	31
%%toc.....	31

Contenido

Parte VII – Dominando las Directivas de Configuración.....	33
%!Target.....	33
%!Options.....	33
%!Encoding.....	34
%!PreProc.....	34
%!PostProc.....	35
%!Style.....	35
Definiendo una Configuración para un Tipo de Archivo Destino Específico.....	36
Detalles de los filtros PreProc y PostProc.....	36
Parte VIII – Mágia Negra.....	39
Insertando Múltiples Líneas con %!PostProc (como reglas CSS).....	39
Creando Contenido "Específico al Tipo Destino" con %!PreProc.....	39
Cambiando Marcas txt2tags con %!PreProc.....	40
Parte IX – História de Txt2tags.....	41
Enero de 1999: Pre-História.....	41
Junio de 1999: Todavía Pre-História.....	41
Agosto de 2000: No más Pre-História.....	41
Mayo 2001: Pythonización e Idea Multi Formato de Destino.....	42
Julio de 2001: séries 0.x: Debut de txt2tags (Lanzamiento Mundial).....	42
Septiembre de 2002: séries 1.x : Crecimiento.....	42
Julio de 2004: séries 2.x : Maduración.....	43

Parte I – Introducción a Txt2tags

Tus Primeras Preguntas

Este capítulo presenta txt2tags, introduciendo a las funcionalidades y objetivos del programa.

¿Que és Txt2tags?

Txt2tags es una herramienta de formateo y conversión de texto. Txt2tags convierte un archivo de texto con algunas marcas a uno de los siguientes tipos de archivo:

- Documento HTML
- Documento XHTML
- Documento SGML
- Documento LaTeX
- Página man de UNIX
- Presentación Magic Point
- Página MoinMoin
- Documento PageMaker 6.0
- Texto plano (sin marcas)

¿Por Qué Debo Usar Txt2tags?

Encontrarás txt2tags muy útil si:

- precisas publicar documentos en diferentes formatos
- precisas mantener documentos actualizados en diferentes formatos
- escribes manuales o documentos técnicos
- no sabes como escribir un documento en un formato específico
- no posees un editor específico para cierto formato
- quieres usar un editor de texto simple para actualizar tus documentos

Pero la motivación principal es:

- economizar tiempo, escribiendo **contenido** sin preocuparse del **formato**

¿Qué ventajas ofrece, comparado con otras herramientas?

Txt2tags se encuentra en constante crecimiento, siguiendo una serie de conceptos basicos:

<i>Archivo fuente legible</i>	Las marcas de txt2tags son muy simples, casi naturales
<i>Documento destino legible</i>	Como el archivo fuente, el código del documento generado es también legible
<i>Consistencia en las marcas</i>	Las marcas txt2tags son únicas, atendiendo todos los tipos de documentos, sin ser confundidas con el contenido
<i>Reglas consistentes</i>	De la misma manera que las marcas, las reglas a ellas aplicadas son uniformes, sin excepciones ni casos especiales..
<i>Estructuras simples</i>	Todas las estructuras soportadas son simples, sin opciones extras o modificadores complejos del comportamiento. Una marca es solo una marca, sin opciones adicionales.
<i>Facil de aprender</i>	Con marcas simples y código fuente legible, presenta una curva de aprendizaje muy amigable.

<i>Buenos ejemplos</i>	Los archivos de ejemplos incluidos en el paquete están basados en documentos reales, son simples, bien estructurados y están escritos en formato txt2tags.
<i>Herramientas valiosas</i>	Los archivos de sintaxis , incluidos en el paquete (para los editores Vim, Emacs, Nano y Kate) ayudan a escribir documentos sin errores sintácticos.
<i>Tres interfaces de usuario</i>	Tiene una interfaz Gráfica Tk , una interfaz Web , que permite usar el programa remotamente o através de una intranet, y una interfaz en línea de comandos para usuarios avanzados y scripts.
<i>Soporta scripts</i>	Con el modo completo de línea de comandos, un usuario avanzado puede automatizar tareas y efectuar post–ediciones en los archivos convertidos.
<i>Descargar y Ejecutar / Multiplataforma</i>	txt2tags es un script Python . No hay necesidad de compilarlo o cargar módulos extra.Puede ser ejecutado sin problemas en máquinas UNIX, Linux,Windows y Macintosh .
<i>Actualizaciones Frecuentes</i>	El programa posee una lista de discusión con usuarios activos que sugieren correcciones y mejoras. El propio autor es un usuario "intensivo" en casa y en su trabajo, por lo que es difícil que el desarrollo pare.

¿ Cuanto cuesta ?

¡ABSOLUTAMENTE NADA!

Txt2tags es un programa gratuito,GPL,open source,dominio público,etc...

Puedes copiarlo,usarlo,modificarlo,venderlo.. etc. Las políticas de software y copyright no se encuentran entre las mayores preocupaciones del autor.

Estructuras de Formato Soportadas por txt2tags

- Encabezamiento (Cabeceras) (título del documento, nombre de autor,fecha)
- Títulos de secciones (numeradas o no)
- Párrafos
- Modificadores de fuentes
 - ◆ Negrita (bold)
 - ◆ Itálica
 - ◆ Subrayada
- Fuente no proporcional (Monoespaciada/Verbatim)
 - ◆ Monoespaciada en un párafo
 - ◆ Línea monoespaciada
 - ◆ Zona en monoespaciado (multilínea)
- Cítas
- Enlaces
 - ◆ URL/ enlaces internet
 - ◆ enlaces e–mail
 - ◆ enlaces locales
 - ◆ enlaces nombrados
- Listas
 - ◆ listas marcadas
 - ◆ listas numeradas
 - ◆ listas de definicion
- Líneas de separación horizontal
- Imágenes (con un alineamiento adecuado)
- Tablas (con o sin borde,con alineamiento , con expansión de columna)
- Marcas especiales para texto raw (que no debe ser tratado)

- Macro especial para la fecha actual (con formato flexible)
- Comentarios (para notas, TODO, FIXME, etc)

Formatos de salida soportados

HTML

Todos sabemos lo que es HTML.

Txt2tag genera documentos HTML limpios, con una buena presentación y cuyo código fuente es legible. No utiliza javascript, frames u otras técnicas fútiles, frecuentemente innecesarias en el formateo de documentos técnicos. Pero, si se desea, puede utilizarse un archivo CSS externo. Txt2tag genera código "*HTML 4.0 Transitional*".

Desde la versión 2.0 de txt2tags el código HTML generado es aprobado al 100% por el [validador w3c](#).

XHTML

Es la nueva generación de HTML con reglas mas estrictas, como la de tener que cerrar todas las marcas que se abran. De esta manera se consigue un código más fácil de tratar y de entender. Para propósitos generales, se puede considerar como HTML. Txt2tag genera código "*XHTML 1.0 Transitional*".

Desde la versión 2.0 de txt2tags el código XHTML generado es aprobado al 100% por el [validador w3c](#).

SGML

Es un formato de documento común que posee aplicaciones de conversión, como [sgmltools](#). A partir de un archivo SGML se pueden generar documentos html, pdf, info, latex, lynx, rtf y xml. Las herramientas como sgml2* también crean índices (TOC) automáticamente, dividiéndolos en secciones y subpáginas (sgml2html)

Txt2tag genera archivos SGML válidos para DTD linuxdoc, listos para ser convertidos mediante una herramienta sgml2* sin archivo de catálogo extra u otras incomodas exigencias de SGML.

LATEX

Es uno de los formatos de documento preferido en los medios académicos y es un formato más potente de lo que uno, de entrada, se imagina. Libros completos, fórmulas complicadas o cualquier texto complejo puede ser escrito en LaTeX. Pero si intentas escribir a mano las marcas requeridas, prepárate para la desesperación.

Txt2tag genera archivos LaTeX listos para usar, realizando todo el complejo trabajo que requiere este formateo, con todas las excepciones. El escritor debe preocuparse sólo por el contenido.

LOUT

Muy parecido a LaTeX, pero con una sintaxis mas fácil, utilizando "@" en lugar de "\" . Su proximidad a la idea de que "todo-es-objeto" da lugar a un marcado mas "sano".

Txt2tag genera archivos Lout listos para su uso, que pueden ser convertidos a PS o PDF utilizando la orden "lout".

MAN

Las páginas man de UNIX resisten el paso del tiempo. Los formatos de documentos vienen y van, pero ellas siguen aquí, inamovibles.

Hay otras herramientas para generar páginas man, pero txt2tag tiene una ventaja: una fuente, múltiples formatos de destino. Así, el contenido de una misma página man puede ser convertido en una página HTML, una presentación Magic Point, etc.

MGP (Magic Point)

Magic Point es una herramienta de presentación bastante útil (tipo: Microsoft PowerPoint), que utiliza un lenguaje de marcado para definir todas las teclas. De esta forma puedes crear presentaciones complejas con vi/emacs/notepad.

Txt2tag genera un archivo .mgp listo para ser utilizado, definiendo todos los encabezamientos necesarios para las definiciones de fuentes y apariencia, así como el soporte a los caracteres acentuados (código ISO-8859).

Observación 1 : los archivos .mgp creados por txt2tag usan las fuentes Type1 de Xfree86. De esta forma no es necesario incluir fuentes TrueType a su presentación.

Observación 2 :: las definiciones de colores para las fuentes son limpias de forma que, incluso con una paleta de colores de sistema pobre, (cómo `startx -- -bpp 8`), la presentación va a tener una buena apariencia.

La clave es : Convierte y Usa. Sin necesidad de posteriores u otros requisitos.

MOIN

¿ No sabes lo que és [MoinMoin](#)? ¡Es un [WikiWiki](#)!

La sintaxis de Moin es un poco liada: tienes que escribir cosas cómo {{{'adding braces and quotes'}}}. Pero txt2tag viene con marcadores simplificados y una solución unificada: una fuente, múltiples formatos destino.

PM6

¿Sabías que Adobe PageMaker 6.0 tiene su propio lenguaje de marcado?. Estilos, tablas de colores, modificadores de fuente y la mayor parte de las funcionalidades accesibles via 'clic' de ratón, también están disponibles en su lenguaje de marcado. Sólo tienes que acceder al menú "Import tagget text", aunque sea sólo por curiosidad, y podrás ver que es un lenguaje de marcas semejante al HTML.

Txt2tag genera todas las marcas y define un encabezamiento extenso y funcional, configurando los estilos de párrafo y formatos. **¡Atención!**: Sin saltos de línea. Cada párrafo debe ser una sola línea.

Nota del Autor: *Todo mi libro sobre [expresiones regulares](#) fue escrito con Vi, convertido al formato PageMaker con txt2tag y enviado a la imprenta.*

TXT

TXT es texto. El único tipo de formato verdadero.

A pesar que las marcas de txt2tag son intuitivas y discretas, puedes eliminarlas convirtiendo el archivo al formato TXT puro.

Los títulos estaran subrayados y el texto es, basicamente, conservado cómo estaba en el archivo fuente.

Estructuras Soportadas en los Archivos finales

Estructura	html	xhtml	sgml	tex	lout	man	mgp	moin	pm6	txt
cabeceras	S	S	S	S	S	S	S	N	N	S
sección título	S	S	S	S	S	S	S	S	S	S
párrafos	S	S	S	S	S	S	S	S	S	S
negrita	S	S	S	S	S	S	S	S	S	-

itálica	S	S	S	S	S	S	S	S	S	–
subrayado	S	S	–	S	S	–	S	S	S	–
f.monoespacio	S	S	S	S	S	–	S	S	S	–
línea verbatim	S	S	S	S	S	S	S	S	S	–
área verbatim	S	S	S	S	S	S	S	S	S	–
área de cita	S	S	S	S	S	S	S	S	S	S
links internet	S	S	S	–	–	–	–	S	–	–
links e–mail	S	S	S	–	–	–	–	S	–	–
links locales	S	S	S	N	N	–	–	S	–	–
links nombrados	S	S	S	–	–	–	–	S	–	–
listas marcadas	S	S	S	S	S	S	S	S	S	S
listas numeradas	S	S	S	S	S	S	S	S	S	S
listas definición	S	S	S	S	S	S	N	N	N	S
línea horizontal	S	S	–	S	S	–	S	S	N	S
imágen	S	S	S	S	S	–	S	S	N	–
tabla	S	S	S	S	N	S	N	S	N	N
Extras	html	xhtml	sgml	tex	lout	man	mgp	moin	pm6	txt
alineación imágen	S	S	N	N	Y	–	S	N	N	–
alin celdas tb	S	S	S	S	N	Y	N	S	N	N
span columna tb	S	S	N	N	N	N	N	N	N	N

Significados

S Soportado

N No soportado (tal vez en futuras versiones....)

– No soportado (no es posible en este tipo de archivo final)

Las tres interfaces de usuario: GUI, Web y Línea de Comandos

Considerando que distintos usuarios pueden tener distintas necesidades y/o disponer de entornos diversos, txt2tags flexibiliza su forma de utilización, presentándose en tres posibles interfaces de usuario, cada una de ellas con sus propias peculiaridades y funcionalidades.

- **Gui:** Escrita en Tk. Aporta a txt2tags un entorno de ventanas y la facilidad del "clic" de ratón.
- **Web:** Escrita en PHP, permite operar remotamente con txt2tags, mediante un navegador, sin necesidad de instalación local
- **Línea de Comandos:** Escrita en Phyton, es el núcleo del programa. Todas las funcionalidades y opciones del programa son accesibles desde la Línea de Comandos.

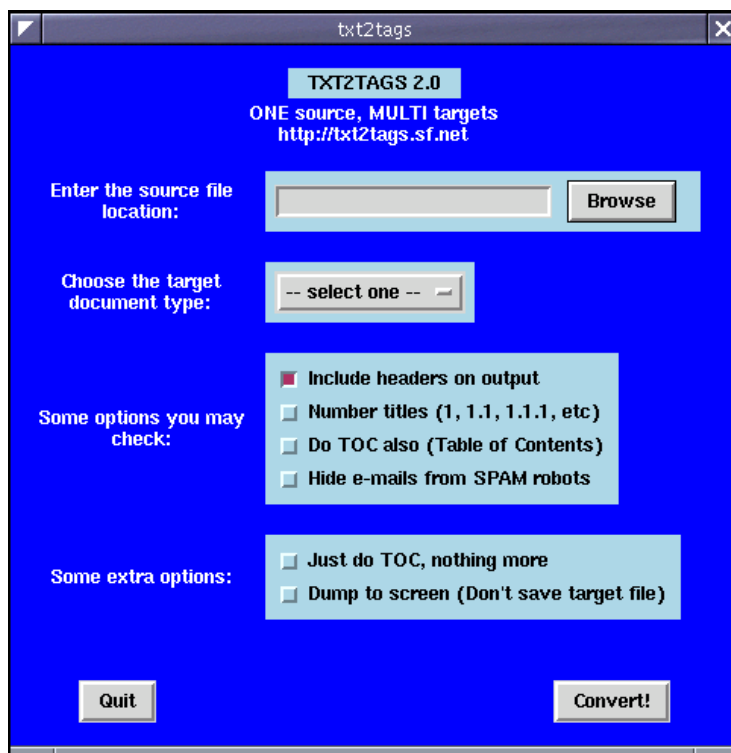
Interfaz Gráfica Tk

Desde su versión 1.0, txt2tags dispone de una interfaz gráfica que funciona en Linux, Mac, Windows y otros sistemas operativos.

Al invocar txt2tags (desde la línea de comandos), el programa detecta automáticamente si el sistema soporta la interfaz gráfica y, en caso afirmativo, la lanza sin necesidad de órdenes adicionales. Se puede forzar el lanzamiento de la interfaz gráfica mediante la opción `--gui`. Si se detectara la falta de algún componente para el funcionamiento de la interfaz gráfica, el programa nos lo advertirá.

Nota: Se necesita el módulo Tkinter. Forma parte de la distribución estandar de Python. De tener instalado éste, debería disponerse del citado módulo.

La interfaz es simple e intuitiva:



1. Localizar el archivo fuente .t2t en el disco.
2. Elegir un tipo de documento de salida.
3. A continuación podrán elegirse una serie de opciones, aunque ninguna de ellas es imprescindible.
4. Finalmente, activando "Convert!", se realiza la conversión del documento.

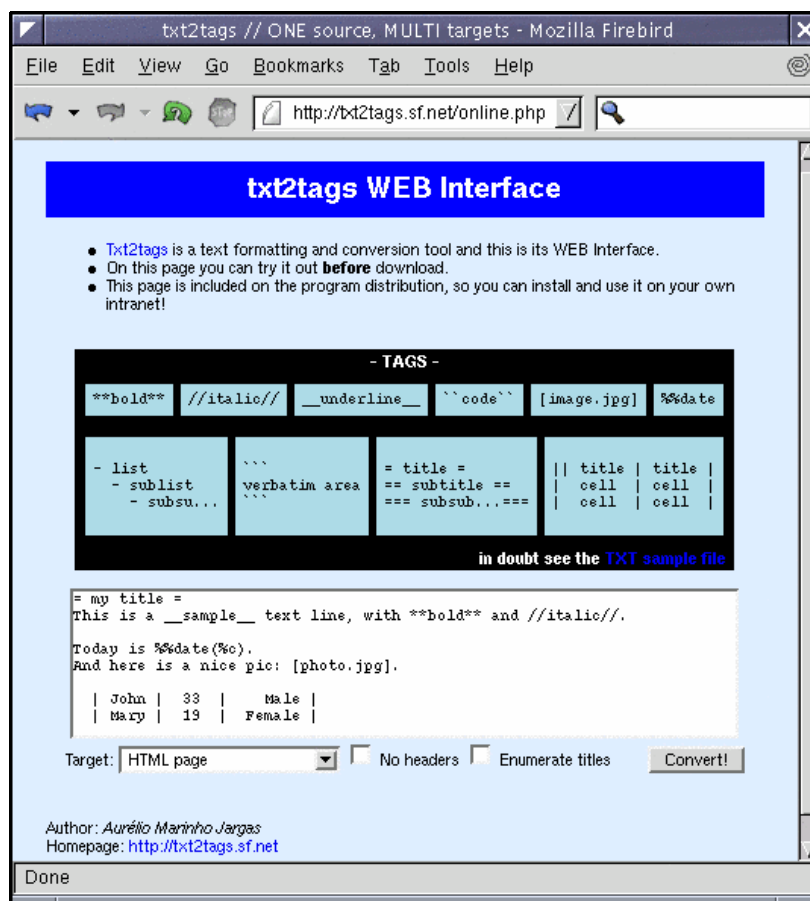
Una de las opciones cuya activación puede ser útil es "*Dump to screen*", (Visualización en pantalla), que nos permite ver el código generado en una ventana distinta a la de trabajo sin que, por el momento, se guarde ningún archivo. Si el código se considera correcto, desactivamos la opción y la conversión se realizará normalmente.

Los colores que vienen por defecto pueden cambiarse en el archivo `~/ .txt2tagsrc` usando los ajustes `%!guicolors` Por ejemplo:

```
%! set my own colors for the graphical interface (bg1, fg1, bg2, fg2)
%!guicolors: blue white brown yellow
```

Interfaz Web

La interfaz web se encuentra disponible en: <http://txt2tags.sf.net/online.php> . Allí puedes usar y probar el programa sin necesidad de instalarlo localmente.



También puede instalarse esta interfaz en la intranet local, evitando así tener que proceder a la instalación de txt2tags en cada una de las computadoras.

Interfaz de Línea de Comandos (Cónsola)

A los usuarios de esta interfaz, la opción `--help` les debe proporcionar salida en pantalla parecida a ésta:

```
Usage: txt2tags [OPTIONS] [infile.t2t ...]

-t, --target          set target document type. currently supported:
                    html, xhtml, sgml, tex, lout, man, mgp, moin, pm6, txt
-i, --infile=FILE    set FILE as the input file name ('-' for STDIN)
-o, --outfile=FILE   set FILE as the output file name ('-' for STDOUT)
-n, --enum-title     enumerate all title lines as 1, 1.1, 1.1.1, etc
-H, --no-headers     suppress header, title and footer contents
--headers            show header, title and footer contents (default ON)
--encoding            set target file encoding (utf-8, iso-8859-1, etc)
--style=FILE         use FILE as the document style (like HTML CSS)
--css-sugar          insert CSS-friendly tags for HTML and XHTML targets
--css-inside        insert CSS file contents inside HTML/XHTML headers
--mask-email        hide email from spam robots. x@y.z turns <x (a) y z>
--toc               add TOC (Table of Contents) to target document
--toc-only          print document TOC and exit
--toc-level=N       set maximum TOC level (depth) to N
--rc                read user config file ~/.txt2tagsrc (default ON)
--gui               invoke Graphical Tk Interface
-q, --quiet         quiet mode, suppress all output (except errors)
-v, --verbose       print informative messages during conversion
-h, --help         print this help information and exit
-V, --version      print program version and exit
--dump-config      print all the config found and exit
```

Turn OFF options:

```
--no-outfile, --no-infile, --no-style, --no-encoding, --no-headers  
--no-toc, --no-toc-only, --no-mask-email, --no-enum-title, --no-rc  
--no-css-sugar, --no-css-inside, --no-quiet
```

Example:

```
txt2tags -t html --toc myfile.t2t
```

By default, converted output is saved to 'infile.<target>'.

Use --outfile to force an output file name.

If input file is '-', reads from STDIN.

If output file is '-', dumps output to STDOUT.

Ejemplos:

Suponiendo que se haya creado un archivo marcado, denominado file.t2t, veamos varias opciones de conversión:

```
Convertir a HTML          $ txt2tags -t html file.t2t  
Usando redireccionamiento $ txt2tags -t html -o - file.t2t > file.html  
.  
Con Índice de Contenidos  $ txt2tags -t html --toc file.t2t  
Con Títulos Numerados    $ txt2tags -t html --toc --enum-title file.t2t  
.  
Vista rápida del Índice   $ txt2tags --toc-only file.t2t  
¿Índice numerado? ¡Si!   $ txt2tags --toc-only --enum-title file.t2t  
.  
..... desde STDIN      $ echo -e "\n**bold**" | txt2tags -t html --no-headers  
-  
Enmascarando los e-mails $ echo -e "\njohn.wayne@farwest.com" | txt2tags -t txt  
--mask-email --no-headers -  
Edición post-conversión  $ txt2tags -t html -o- file.t2t | sed "s/<BODY  
.*/<BODY BGCOLOR=green>/" > file.html
```

Nota

Desde la versión 1.6 es posible realizar un tratamiento pre y post conversión mediante los filtros %!preproc y %!postproc .

Parte II: – OK!, Quiero utilizar el programa. ¿Que debo hacer ?

Sencillamente, descargar el programa y hacer que corra en tu máquina :-). Veamos como hacerlo.

Obtener e instalar Python

En primer lugar, deberemos instalar el interprete Python en nuestro sistema. Si ya lo tenemos instalado, podemos saltarnos este apartado.

Python es uno de los mejores lenguajes de programación. Funciona en Linux, Mac, Windows y otros Sistemas Operativos. Puede descargarse de la web de [Python](http://python.org). Las instrucciones para su instalación se encuentran disponibles en este mismo site. Txt2tags funciona mejor con versiones de Python 1.5 o superiores.

Si no estais seguros de tener Python instalado en vuestro sistema, abrid una consola (tty, xterm, MSDOS..) y lanzad el comando `python`. Si no está instalado, vuestro sistema os lo advertirá

Obtener Txt2tags

El sitio oficial de la distribución de txt2tags es: <http://txt2tags.sf.net/src> .

Todos los archivos son del tipo "tarball" (tar.gz,.tgz), que pueden ser descomprimidos por la mayoría de los programas de descompresión (incluido Winzip).

Debemos procurar obtener la **última versión** (la mas reciente o el número de versión mas alto). Las versiones precedentes se conservan en el repositorio por razones puramente históricas.

Instalar txt2tags

Como todos los scripts Python, txt2tags apenas precisa de instalación.

El único archivo necesario para el programa es el script txt2tags. Los demás archivos tar.gz (.tgz) son la documentación, las utilidades y los archivos de ejemplo.

El medio mas seguro de iniciar txt2tags es "llamarlo" desde Python.

```
prompt$ python txt2tags
```

Si deseais instalar txt2tags como programa en vuestro sistema, copiad (o cread un enlace) el script txt2tags en un directorio referenciado por la variable \$PATH y aseguraos de que el sistema pueda lanzar el programa.

UNIX/Linux

Asegurémonos de que el script es ejecutable : `chmod +x txt2tags` y, seguidamente lo copiamos en un directorio referenciado por \$PATH: `cp txt2tags /usr/local/bin`

Windows

Renombrar el script añadiéndole la extensión .py : `ren txt2tags txt2tags.py` y copiarlo en un directorio referenciado en \$PATH: `copy txt2tags.py C:\WINNT`

Después de esto, si deseais utilizar el entorno gráfico, podreis crear un icono en vuestro escritorio.

Paquetes especiales para los usuarios de Windows

Existen dos distribuciones .EXE de txt2tags, que instalan el programa en maquinás Windows con solo unos cuantos "clicks":

- La primera es el simple script txt2tags para los que tengan instalado Python.
- La segunda de estas dos distribuciones es la "versión completa", con la cuál no se precisa tener instalado Python en el sistema (Incluye una versión "ligera" de Python).

Se pueden obtener en *Txt2tags-Win*: <http://txt2tags-win.sf.net>

Instalación de los Archivos de Sintaxis para los Editores de Texto

Con txt2tags se proporcionan una serie de archivos de sintaxis prácticos, utilizables con los siguientes editores:

- Vim (www.vim.org)
- Emacs (www.emacs.org)
- Nano (www.nano-editor.org)
- Kate (<http://kate.kde.org>)

Estos archivos sintácticos contienen todas las reglas y marcas que utiliza txt2tags, ayudádo al usuario a escribir los documentos sintacticamente correctos. Las marcas se visualizan en color, así podremos darnos cuenta de si escribimos correctamente.



```
aurelio@ibook:~/txt2tags/src
TXT2TAGS SAMPLE
Aurelio Marinho Jargas
%%date

%encoding: iso-8859-1

= Introduction =

Welcome to the txt2tags sample file.

Here you have examples and a brief explanation of all
marks.

The first 3 lines of the this file are used as headers,
on the following format:
```
line1: document title
line2: author name, email
line3: date, version
```

Lines with balanced equal signs = around are titles.

% a secret comment!
%T000 link to program homepage http://txt2tags.sf.net

= Fonts and Beautifiers =

We have two sets of fonts:
samples/sample.t2t 1,1 Top
:wq
```

Ejemplo de archivo editado en Vim

Cada editor tiene un procedimiento de instalación específico. Consulta el encabezamiento del archivo de sintaxis y la documentación del editor.

Parte III – Escribiendo y convirtiendo vuestro primer documento.

Probar las herramientas

Para poder hacer vuestra primera conversión necesitareis tres cosas: txt2tags, un editor y un navegador web.

1. Comprobad que txt2tags funciona correctamente en vuestra máquina.
 - ♦ **Interfaz de Línea de Comandos:** Lanzad el comando txt2tags desde la línea de comandos. Deberá generarse una salida con un mensaje similar a "Missing input file". Si no ocurre eso, probad con: "python /ruta/hacia/txt2tags" o con: "/ruta/hacia/python /ruta/hacia/txt2tags" en el caso de que python no esté en vuestro PATH.
 - ♦ **Interfaz GUI:** Si lanzamos txt2tags desde la línea de comandos, se iniciará con el sistema gráfico. En el caso de que hayamos creado un icono en el escritorio, "clicar" en el.
2. Abrid un editor de texto que conozcáis bien, puede ser cualquiera, desde Vim a MSWord u Openoffice. Cread un nuevo documento, que será vuestro primer documento txt2tags.
3. Lanzad vuestro browser preferido para poder ver los resultados de la conversión en una pagina HTML.

Escribir el encabezamiento del documento

1. Id al editor de texto y teclead una primera línea , por ejemplo: *Mi Primer Documento*
2. En una segunda línea teclead un sub-título, por ejemplo: *txt2tags*
3. Después, en una tercera línea teclead, por ejemplo, una fecha: *Domingo 17 Julio 2005*

Así, dispondremos de un documento con el siguiente contenido:

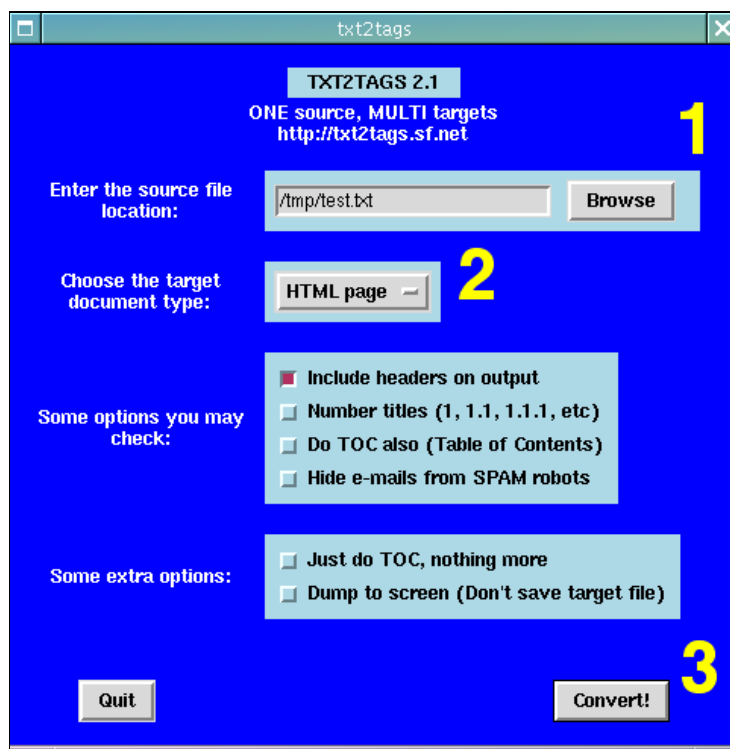
```
Mi primer documento
txt2tags
Domingo 17 Julio 2005
```

Aunque ésto sea sólo una parte del documento, podemos convertirlo y comprobar el resultado. Guardad el documento con el nombre `test.txt` en un directorio desde donde, más tarde, podamos recuperarlo (en los siguientes ejemplos el documento se ha guardado en el directorio `/tmp`).

La Primera Conversión : La Interfaz Gráfica.

Si utilizais la interfaz de línea de comandos, pasad a la siguiente sección.

Si utilizais la interfaz gráfica, proceded tal como a continuación se indica:



1. Clicar en el botón "Browser", navegad hasta el directorio en el que guardasteis el documento test.tx y seleccionarlo.
2. En el apartado "Choose the target document type" (Elija el tipo de documento de salida) escoged : "HTML page"
3. Clicar el boton "Convert!" (Convertir)



Aparecerá un diálogo que nos informa que la conversión se ha realizado con éxito. Notad que el archivo HTML generado ha sido guardado en el mismo directorio que el archivo test.txt, pero con la extensión .html

La Primera Conversión: Interface en Línea de Comandos

Si utilizais la interfaz gráfica, id a la sección anterior.

Si utilizais la interfaz de Línea de Comandos, id al directorio en el que está guardado test.txt y teclead el siguiente comando:

```
txt2tags --target html test.txt
```

Observad que hay un espacio en blanco entre cada una de las partes del comando, pero no en `--target`. Dicha opción es seguida por `html`, que indica al programa el formato al que deseamos convertir el archivo de texto. La última parte de la orden es el nombre del archivo de texto que

deseamos convertir.

Si la conversión se realiza con éxito, el resultado es guardado como `test.html` y el programa nos dará un mensaje del tipo: "`txt2tags wrote test.html`". En caso contrario, os indicará el error cometido. Verificadlo con atención.

Esto es una muestra de lo que podeis ver en la pantalla:

```
prompt$ txt2tags --target html test.txt
txt2tags wrote test.html
prompt$
```

Comprobar el resultado

Abrir el archivo `test.html` con el navegador y comprobar el resultado de la conversión.



¡ Bien ! Os habeis limitado a teclear tres simples líneas de texto y `txt2tags` ha realizado todo el trabajo necesario para configurar el encabezamiento de la pagina HTML. El alineamiento del texto, los márgenes, los espaciados, la apariencia, etc. Podemos observar como el título principal del documento aparece también en el borde superior del navegador (o en el título de la pestaña, en el caso de que usemos uno que aporte esta función).

Vosotros escribis el texto, txt2tags hace el resto :-)

Truco: podeis utilizar CSS en las páginas HTML generadas por `txt2tags`. De esta forma la apariencia de la página es configurable al 100%.

Escribir el cuerpo del documento

Volvamos al editor de texto, ya que la siguiente etapa consistirá en escribir el cuerpo del documento. Podeis escribirlo en texto plano, el mismo que usais en los e-mail. Podreis observar como `txt2tags` reconoce automaticamente los párrafos y las listas de elementos, sin necesidad de "marcarlos".

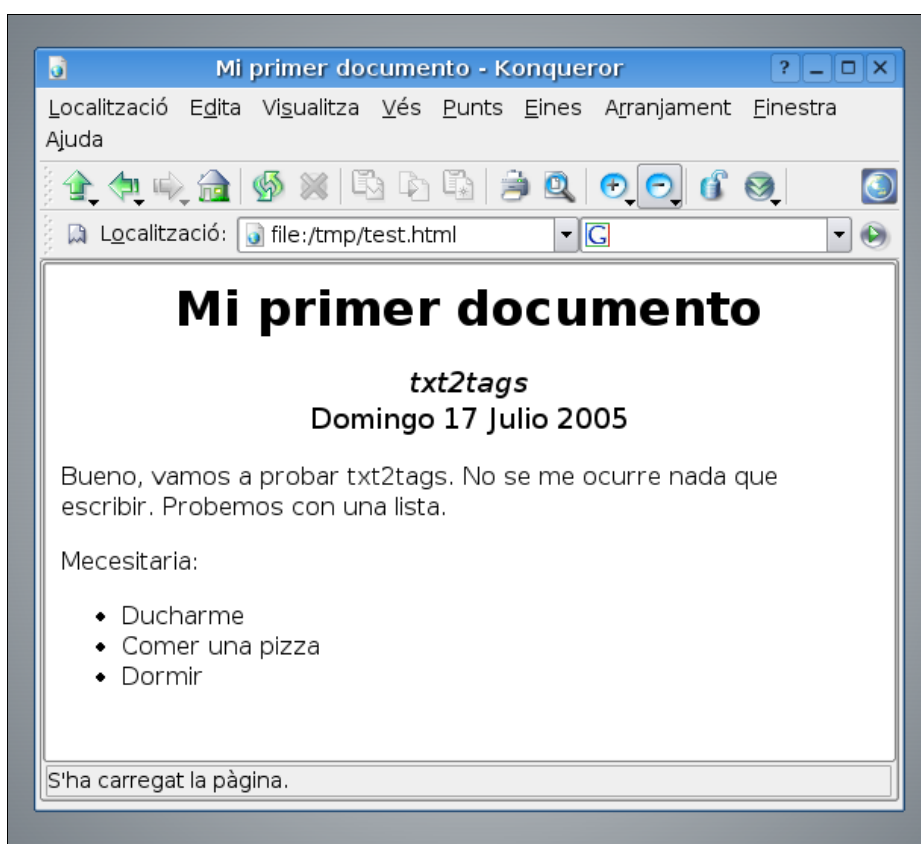
Guardad de nuevo el documento y proceded a su conversión, tal como se indicó en la anterior sección. Comprobad el resultado. Éste es el ciclo de desarrollo de documentos con `txt2tags`. No

necesitais más que centraros en el contenido del documento, obteniendo el documento final mas rápidamente que con otros editores. Sin interminables "clics" de ratón, ni menús, ni ventanas ni distracciones.

Observad el contenido del archivo de texto plano `test.txt` y comparémoslo con el archivo HTML generado:

```
Mi primer documento
txt2tags
Domingo 17 Julio 2005
Bueno, vamos a probar txt2tags.
No se me ocurre nada que escribir. Probemos con una lista.
```

```
Necesitaria:
- Ducharme
- Comer una pizza
- Dormir
```



Con txt2tags es posible escribir una página web entera sin ningún conocimiento de HTML. No necesitareis intartar ninguna marca. Además, el mismo archivo de texto plano puede convertirse en cada uno de los demás formatos soportados por txt2tags.

Además del texto plano, txt2tags proporciona una serie de marcas muy simples que pueden utilizarse cuando se tenga necesidad de otros formatos o estructuras, como puede ser el texto en negrita, cursiva o itálica, los títulos, las imágenes, las tablas y otros. Un ejemplo rapido:

```
**comienza en negrita** y == los signos igual para el título ==.
```

Para aprender a utilizar estas marcas consultad el archivo [Txt2tags Markup Demo](#).

Parte IV – Dominando los Conceptos de Txt2tags

Las Áreas de un Documento .t2t

Los archivos marcados por txt2tags se dividen en 3 áreas. Cada una de estas tres áreas tiene sus propias reglas y finalidades. Estas áreas son:

ENCABEZAMIENTO

Lugar para el título del documento, nombre del autor, versión y fecha. (opcional)

CONFIGURACIÓN

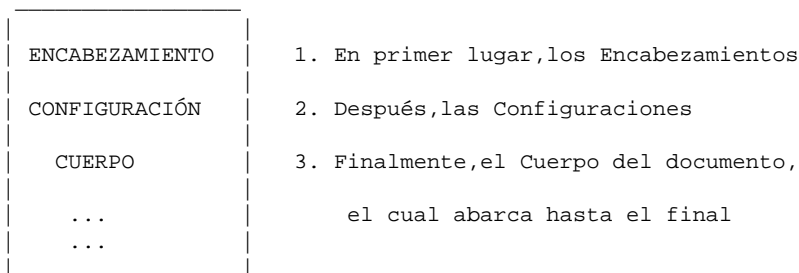
Lugar para la configuración general del documento. (opcional)

CUERPO

Lugar en el que se coloca el contenido del documento. (obligatoria)

Como se ha podido observar, solo el área del *CUERPO* es obligatoria.

Cada área esta regulada por reglas específicas, que se detallarán en el siguiente capítulo. Por ahora, vamos a quedarnos con una representación gráfica de las áreas de un documento txt2tags:



En resumen, así es como se definen las tres áreas:

- Encabezamiento** Las primeras 3 líneas del archivo. O la primera línea en blanco, cuando no se quiere un Encabezamiento.
- Configuración** Desde la 4ª línea (o desde la 2ª, en el caso que no se contemple Encabezamiento) hasta donde comience el Cuerpo del Documento..
- Cuerpo** Comienza con la primera línea de texto válido (no "comentado" o que no sea una configuración) después del Encabezamiento.

Veamos un ejemplo:

```
My nice doc Title
Mr. John Doe
Last Updated: %%mtime(%c)

%! Target : html
%! Style : fancy.css
%! Encoding: iso-8859-1
%! Options : --toc --enum-title

Hi! This is my test document.
Its content will end here.
```

Área de Encabezamiento

Localización:

- Posición fija: Las **tres primeras líneas** de un archivo. Eso es todo :-)
- Posición fija: La **primera línea** de un archivo, si se trata de una línea en blanco. Esto significa "Encabezamiento vacío".

El Área de Encabezamiento es la única que tiene una posición fija, determinada por líneas. Se encuentra siempre en las tres primeras líneas del archivo fuente.

El contenido de estas líneas no está regulado, es libre, aunque se recomienda que se siga la siguiente pauta:

- *línea 1*: título del documento
- *línea 2*: nombre del autor y/o email
- *línea 3*: fecha del documento y/o versión (un buen lugar para %%date)

Retengamos que las tres primeras líneas del documento fuente serán las tres primeras líneas del documento final, separadas y diferenciadas del resto del contenido del documento (p.e., caracteres más grandes, en negrita). Si es posible la paginación, encontraremos el encabezamiento sólo y centrado en la primera página.

Menos (o Ninguna) líneas de Encabezamiento

Tal vez deseemos menos líneas de encabezamiento. Por ejemplo, solo deseamos especificar el título y la fecha.

Para ello bastará con dejar la 2ª y/o la 3ª línea en blanco. Esta/s líneas en blanco no serán incluidas en el documento final. Pero no debemos olvidar que, aún en blanco, estas líneas forman parte del Área de Encabezamiento del documento y que el Cuerpo del mismo comenzará **a partir de** la tercera línea.

La primera línea (normalmente, el título) es la única línea requerida en el Área de Encabezamiento. Recordemos que la primera línea en blanco indica **Encabezamiento vacío**, lo que origina que el Área de Encabezamiento se inicie a partir de la 2ª línea.

No configurar un encabezamiento puede sernos de utilidad en el caso de querer insertar un encabezamiento personalizado después de la conversión. Para éste tipo de operación generalmente se requiere la opción de línea de comandos `--no-headers`.

En resumen: Los Encabezamientos son posiciones, no contenidos.

Si colocamos un texto en la primera línea del archivo fuente, aparecerá en la primera línea del documento convertido. Lo mismo para las 2ª y 3ª líneas.

Área de Configuración

Localización:

- Comienza justo después del Área de Encabezamiento
 - ♦ Comienza a partir de la 4ª línea si se especifican encabezamientos
 - ♦ Comienza a partir de la 2ª línea en el caso de **Encabezamiento vacío**.
- Finaliza donde comienza el Área del Cuerpo del documento
 - ♦ Finaliza en una línea en blanco, en una línea que sea un comentario o en una línea

que no sea de ajuste/configuración.

El Área de Configuración es opcional. Cualquiera puede generar documentos txt2tags sin siquiera conocer la existencia de dicha área, pero el usuario experimentado difícilmente podrá prescindir de la ventaja y el control que dicha área le proporcionará.

El Área de Configuración nos servirá para definir en ella configuraciones que afecten al comportamiento del programa, evitando así tener que escribir dichas opciones en la línea de comandos en el momento de realizar la conversión.

Para más información, consultar la Sección: [Configuraciones \(Settings section\)](#).

Área del Cuerpo del Documento

Localización:

- Comienza con la primera línea válida de texto del archivo.
 - ◆ Las líneas que contienen ajustes o comentarios y las líneas en blanco no se consideran líneas válidas de texto
- Finaliza al final del archivo (EOF)

Consideraremos Cuerpo todo lo que no sea ni Encabezamiento ni Configuración.

El Cuerpo está constituido por el contenido del documento y todas aquellas estructuras y formatos que txt2tags puede reconocer. En el Cuerpo pueden incluirse comentarios para los *TODOs* y notas personales.

Puede utilizarse la opción en línea de comandos `--no-headers` para convertir solo el Cuerpo del documento, suprimiendo los Encabezamientos. Esto es práctico cuando se tienen Encabezamientos personalizados en un archivo aparte, que se concatenará seguidamente con el documento que estemos procesando.

Configuraciones: ¿Cómo Configurar algo?. ¿Cuál es la sintaxis?

Las líneas de configuración son líneas de comentario especiales, situadas en el Área de Configuración, que afectan al proceso de conversión. Su sintaxis, tan simple como la atribución de valor a una variable, está compuesta por una palabra-clave y un valor, separados la una del otro por el separador canónico dos-puntos ":" . Su sintaxis es:

%! palabra-clave : valor

Lista de claves válidas:

Clave	Descripción
Target	Define el formato al cuál debe ser convertido el documento.
Options	Es útil para especificar las opciones de línea de comando para el archivo fuente. El formato de las opciones es el mismo que el utilizado en la línea de comandos. Usando esta opción junto con %!target permite convertir el documento simplemente con la orden: <code>txt2tags file.t2t</code>
Style	Sólo es soprtado por el documento destino HTML, para definir una hoja de estilo (CSS)
Encoding	Es una configuración importante en documentos que contienen acentos y caracteres no ASCII, ya que permite personalizar el Conjunto de Caracteres (Codificación de

	Carácteres).
PreProc	Es un filtro. Define reglas del tipo "busca y sustituye" que serán aplicadas al archivo original antes de cualquier análisis por parte de txt2tags.
PostProc	Es un filtro. Define reglas del tipo "busca y sustituye" que serán aplicadas al archivo destino después de todo el procesamiento por parte de txt2tags.

Ejemplo:

```

%! Target   : html
%! Options  : --toc --toc-level 3
%! Style    : fancy.css
%! Encoding : iso-8859-1
%! PreProc  : "AMJ"          "Aurelio Marinho Jargas"
%! PostProc : '<BODY.*?>'   '<BODY bgcolor="yellow">'
    
```

Algunas reglas sobre las Configuraciones

- Las configuraciones son válidas solo dentro del Área de Configuración. Serán tratadas como comentarios simples si se encuentran en el Cuerpo del documento.
- Si la misma palabra-clave aparece más de una vez en el Área de Configuración, solo se utilizará la última. Excepciones: Options, PreProc y PostProc, que son acumulativas.
- Una línea de configuración con una palabra-clave inválida será considerada un comentario.
- Las opciones de configuración situadas en el Área de Configuración tienen preferencia dentro del archivo txt2tags, pero no sobre las opciones en Línea de Comando.

Opciones en Línea de Comando

El modo más rápido de cambiar el comportamiento por defecto es utilizando las opciones de la línea de comando. Esto es sólo posible cuando se utiliza la citada interfaz, no cuando se utilizan las interfaces Web o GUI.

El programa acepta un juego de opciones predefinidas. Una opción es un guión ("-") seguido de una letra o dos guiones ("--") seguidos de una o más palabras, como `-t` y `--target`. A propósito de la opción "target": resulta ser la única opción obligatoria. Todas las demás opciones son eso, opcionales. :-)

Las opciones más utilizadas son `--outfile` para activar la generación automática de índices de materias y `--encoding` que nos permite escoger el tipo de codificación de caracteres. La mayoría de las opciones pueden ser desactivadas precediéndolas de "no-", por ejemplo: `--no-encoding` and `--no-toc`.

Pueden configurarse las opciones que se deseen utilizar en un determinado archivo origen, en el Área de Configuración del mismo archivo, mediante la regla de configuración `!options`. Esto nos evita el tecleo de las opciones en la línea de comando. Por ejemplo: `!options: --toc -o mydoc.html`. La excepción es la especificación del formato del archivo final, que tiene su propia sintaxis: `!target: html`.

La opción `--help` nos proporcionará una lista completa de las opciones disponibles.

Archivo de Configuración del Usuario (RC)

El archivo de configuración del usuario (llamado también archivo RC) es el lugar donde se guardan las reglas de las configuraciones que pueden ser compartidas por TODOS los archivos. Si utilizas las mismas reglas de configuración en todos los archivos .t2t que escribais, incluídlas en el archivo RC y

podrán ser utilizadas globalmente, tanto por los existentes como por futuros archivos fuente.

El emplazamiento por defecto de este archivo depende del sistema que se esté utilizando. De todas maneras, dicho emplazamiento puede ser modificado por el usuario utilizando una variable de entorno.

	RC file location
Windows:	%HOMEPATH%_t2trc
Linux y otros:	\$HOME/.txt2tagsrc
Definido por usuario:	T2TCONFIG env var

El formato de las reglas de configuración es exactamente el mismo que el utilizado en el Área de Configuración de los archivos .t2t. Podemos encontrar un ejemplo de archivo RC en el "tarball" `doc/txt2tagsrc`

Ejemplo:

```
% my configs

%% Always use CSS-friendly tags in HTML
%!options(html): --css-sugar

%% Change the default TOC depth for all targets
%!options: --toc-level 4

%% Set the default encoding for all documents
%!options: --encoding iso-8859-1
```

Toda línea no vacía, un comentario o una línea de configuración válida puede generar un error durante la ejecución de txt2tags. Sed muy cautelosos a la hora de editar el archivo RC.

Txt2tags aplica automáticamente el archivo RC a todo archivo fuente que convierta. Si se desea desactivar el comportamiento para un determinado archivo, utilizad la opción `--no-rc`.

Orden de Aplicación de las Configuraciones y Precedencia

Hay tres formas de indicar a txt2tags cuales opciones y configuraciones utilizar y el orden que debe aplicarlas:

1. El archivo de usuario RC (`.txt2tagsrc`)
2. Una Área de Configuración en el documento fuente
3. Las opciones de línea de comando

En primer lugar, el programa lee las configuraciones del archivo RC. Si no encuentra otras, las aplica. Si encuentra configuraciones en el Área de Configuración del documento fuente, las lee, sobrescribiendo las configuraciones de RC. Finalmente, si encuentra opciones en la línea de comando, las lee, sobrescribiendo a cualquiera de las otras dos anteriores.

Así, si el "encoding" del documento fuera definido en las tres opciones, se aplicarían las opciones definidas en la línea de comando.

El Comando `%!include`

Desde la versión 1.7, es posible la utilización del comando `include` para insertar el contenido de un archivo externo en el Cuerpo del documento original.

`include` es un comando, no una configuración. Es solo válido cuando se incluye en el *Área del Cuerpo del documento*.

`include` es útil para dividir un documento grande en varios archivos menores (p.e. en capítulos) o para incluir todo el contenido de un archivo externo cualquiera en el documento fuente.

Ejemplo:

```
My first book
Dr. John Doe
1st Edition

%!include: intro.t2t
%!include: chapter1.t2t
%!include: chapter2.t2t
...
%!include: chapter9.t2t
%!include: ending.t2t
```

Como puede verse, basta con indicar el nombre del archivo después del comando `include`. Las especificaciones opcionales del archivo final también son soportadas por el comando:

```
%!include(html): file.t2t
```

Debemos tener presente que `include` insertará en el Área del Cuerpo el archivo `.t2t`, ignorando su encabezamiento y sus configuraciones. Esto nos permitirá convertir el documento incluido, separada o conjuntamente con el documento principal.

Podemos considerar tres variantes de inclusión:

- Verbatim
- Raw
- Tagged

Verbatim : El texto es incluido preservando sus espacios y formato original, como si estuviese marcado por VERB. Para incluir un archivo de esta manera, basta con incluir su nombre entre (``):

```
%!include: ``/etc/fstab``
```

Raw : El texto es incluido "como esta". No busca las marcas `txt2tags`, ni las interpreta. Es como si el texto estuviese entre marcadores (" ") de área protegida. Para incluir un archivo de esta manera basta con colocar su nombre entre dobles-comillas (" "):

```
%!include: ""boom_text.txt""
```

Tagged : El texto es pasado directamente al documento resultante, sin ningún tratamiento por parte de `txt2tags`. Mediante este procedimiento podemos incluir en el documento final textos marcados o con estructuras complejas no soportadas por `txt2tags`. Otra aplicación sería la de incluir un encabezamiento o un pie de página en todos los documentos. Para aplicar esta variante de inclusión debemos encerrar el nombre del archivo entre dobles-simples-comillas (" ") no entre dobles-comillas ("") como en el anterior caso.

```
%!include(html): 'footer.html'
```

Como el contenido de `cabecera.html` contiene tags, es conveniente especificar el destino (`html`) para evitar problemas..

El Comando %!includeconf

`includeconf` se utiliza para incluir la configuración de un archivo externo en el archivo corriente. Es sólo válido cuando se incluye en el Área de Configuración.

Es práctico cuando se desea tener la misma configuración para múltiples archivos, permitiéndonos centralizar su gestión. En el archivo en el que se quiera incluir la configuración global, incluir una llamada `includeconf`.

Ejemplo:

```
Mi Primer Documento
txt2tags
Domingo 17 Julio 2005

%!includeconf: config.t2t

Este es un ejemplo de includeconf
```

El formato del archivo incluido es el mismo que en el [archivo RC](#).

Part V – Dominando las Marcas

Resúmen de las Marcas:

Básicas	Modificadores
<i>Encabezamiento</i>	primeras 3 líneas	<i>Negrita</i>	**palabras**
<i>Título</i>	= palabras =	<i>Itálica</i>	//palabras//
<i>Título numerado</i>	+ palabras +	<i>Subrayado</i>	__palabras__
<i>Párrafo</i>	palabras	<i>Monospaciado</i>	`palabras`
Bloque de Texto	Otras
<i>Cita</i>	<TAB>palabras	<i>Línea de separación</i>	-----...
<i>Lista</i>	- palabras	<i>Línea gruesa</i>	=====...
<i>Lista numerada</i>	+ palabras	<i>Enlace</i>	[etiqueta url]
<i>Lista definición</i>	: palabras	<i>Imágen</i>	[filename.jpg]
<i>Línea Verbatim</i>	``` palabras	<i>Comentario</i>	% comentarios
<i>Zona Verbatim</i>	``` \n líneas \n ```	<i>Texto Raw</i>	""palabras""
<i>Línea Raw</i>	"""" palabras	<i>Tabla</i>	celda1 celda2 celda3...
<i>Zona Raw</i>	"""" \n líneas \n """"	<i>Ancla</i>	= titulo =[ancla]

Reglas Generales:

- **Encabezamiento:** son las tres primeras líneas del documento, las marcas no son interpretadas.
- **Títulos:** son caracteres "=" or "+", equilibrados alrededor del texto del título. A mayor número de caracteres, mayor "profundidad" de título (título, sub-título, sub-sub-título, etc).
- **Modificadores de Fuentes:** no aceptan espacios entre las marcas y su contenido.
- **Comentario:** la marca "%" debe situarse al comienzo de la línea (primera columna).
- **Imágenes:** el nombre del archivo de imagen debe acabar en GIF, JPG, PNG o similar.
- Las únicas marcas **multilinea** son las de las zonas verbatim y raw.
- En las zonas verbatim y raw, ninguna marca es **interpretada**.
- las **Líneas de Separación/Líneas Gruesas** deben tener, al menos, 20 caracteres.
- El **(des) anidamiento** de las Citas y Listas viene determinado por la indentación.
- La **fila de encabezamiento de una Tabla** se define con dos barras verticales || al principio de la fila.

Encabezamiento

- **Descripción:** Identifica las cabeceras del documento.
- **Propiedades:** Multilinea, FreeSpaces, !Align, !Nesting
- **Contiene:** Macros
- **Sintaxis:**
 - ♦ Las tres primeras líneas del documento fuente.
 - ♦ Para no especificar encabezamiento, dejar la primera línea blanco
 - ♦ Dejar en blanco la segunda y/o tercera línea para eliminar otras partes del encabezamiento.
- **Detalles:**
 - ♦ Las marcas no son interpretadas.
 - ♦ Las tres primeras líneas del documento origen serán las tres primeras del documento final en el que aparecerá resaltadas. Si la paginación es soportada, pueden aparecer

aisladas en la primera página del documento final.

- ◆ El contenido del encabezamiento puede ser cualquier cosa, sin requisito alguno de formato. De todas formas, se recomienda seguir la siguiente estructura:
 - ◇ Línea 1: Título del documento
 - ◇ Línea 2: Nombre del autor y/o e-mail
 - ◇ Línea 3: Fecha del documento y/o versión (buen lugar para %%mtime)

Título y Título Numerado

- **Descripción:** Identifica un título de sección, numerado o no.
- **Propiedades:** !Multiline, FreeSpaces, !Align, !Nesting
- **Contenido:** Raw
- **Sintaxis:**
 - ◆ Para títulos numerados, sustituir "=" por "+" en las siguientes reglas.
 - ◆ Equilibrio de signos alrededor de las palabras: = asi, si ===así, no =
 - ◆ Mas signos,mas sub-niveles : =título= ==subtítulo== ===sub-subtitulo===
 - ◆ Hay un máximo de 5 niveles =====como esto=====
 - ◆ Los signos no equilibrados no dan lugar a un título : =no es titulo===
 - ◆ Se autorizan espacios dentro las marcas: = como esto =
 - ◆ Los títulos pueden tener anclas: =título=[ancla]. Para crear un ancla, crear un [enlace local#ancla]
- **Detalles:**
 - ◆ Las marcas no son interpretadas.
 - ◆ Las macros no son interpretadas.

Párrafo

- **Descripción:** Identifica un párrafo de texto
- **Propiedades:** Multiline, FreeSpaces, !Align, !Nesting
- **Contenido:** Macros, Modificadores de Fuentes, Raw, Enlaces, Imágenes, Comentarios
- **Sintaxis:**
 - ◆ Los párrafos son grupos de líneas delimitados por líneas vacías.
 - ◆ Otros bloques como las listas, las citas, las tablas y las zonas verbatim también cierran párrafos.

Comentario

- **Descripción:** Usados para insertar un texto que no formará parte del documento final.
- **Propiedades:** !Multiline, !FreeSpaces, !Align, !Nesting
- **Contenido:** --
- **Sintaxis:**
 - ◆ Una línea con un carácter "%" en la primera columna: % como ésto.
 - ◆ Sin espacio delante
- **Detalles:**
 - ◆ Como comentarios, no se muestran en el texto convertido.
 - ◆ No son un bloque: cada línea de comentario debe comenzar con "%"
 - ◆ Útiles para TODOS, llamadas FIXME y notas del editor

Modificadores de Fuentes: Negrita, Itálica y Subrayado

- **Descripción:** Utilizado para insertar texto en negrita/itálica/subrayado, en un párrafo, una tabla, una lista o una cita

- **Propiedades:** !Multiline, !FreeSpaces, !Align, Nesting
- **Contenidos:** Macros, Modificadores de Fuentes, Raw, Enlaces, Imágenes
- **Sintaxis:**
 - ♦ Dos arceriscos alrededor para la negrita: `**como ésto**`
 - ♦ Dos barras alrededor para la itálica: `//como ésto//`
 - ♦ Dos guiones–bajos alrededor para el subrayado: `__como ésto__`
 - ♦ No debe haber espacio entra las marcas y el contenido: `** esto és inválido`
`**`
- **Detalles:**
 - ♦ Todos los modificadores de fuentes deben estar sobre una misma línea del archivo original, no debe haber un salto de línea
 - ♦ Las macros deben emplazarse en el interior de los modificadores de fuentes:
`**%date**`
 - ♦ Pueden anidarse : `**__como__//aqui//**`

Fuente no Proporcional (Monoespaciada)

- **Descripción:** Utilizado para insertar un texto en fuente no proporcional en un párrafo, una tabla o una cita
- **Propiedades:** !Multiline, !FreeSpaces, !Align, !Nesting
- **Contenido:** --
- **Sintaxis:**
 - ♦ Dos apóstrofes inversos: `` `como aqui` ``
 - ♦ No debe haber espacio entre las marcas y el contenido: `` ` esto és inválido`
`` ``
- **Detalles:**
 - ♦ Las marcas no son interpretadas
 - ♦ Las macros no son interpretadas
 - ♦ Todo el texto en fuente no proporcional debe estar en una sóla línea del archivo fuente, no se admiten saltos de línea
 - ♦ En algunos archivos convertidos, los espacios internos son mantenidos, en otros, los espacios múltiples son reemplazados por espacios sencillos
 - ♦ Se puede crear un texto en negrita en fuente no proporcional insertándolo en las marcas de negrita : `**` `negrita` `**`. Lo mismo para los otros dos modificadores, itálica y subrayado: `//` `italica` `// y __` `subrayado` `__`

Línea Verbatim, Zona Verbatim

- **Descripción:** Usado para insertar código–programa u otro texto pre–formateado en un párrafo, tabla, lista o cita, preservando los espacios, los saltos de línea y usando fuente no proporcional (monoespaciada)
- **Propiedades:** Multiline, !FreeSpaces, !Align, !Nesting
- **Contenido:** --
- **Sintaxis: Línea Verbatim:**
 - ♦ Una línea que comienza exactamente con 3 apóstrofes inversos, seguidos de un espacio y seguido de texto: `` ` ` como ésto`
 - ♦ Los apóstrofes inversos deben estar en la primera columna, sin espacios delante
- **Sintaxis: Zona Verbatim:**
 - ♦ Una línea que comienza exactamente con 3 apóstrofes inversos `` ` `` seguidos de líneas de texto, seguidas de otra línea con exactamente 3 apóstrofes inversos `` ` ``.
 - ♦ No está permitido ningun espacio antes y después de las marcas
- **Detalles:**
 - ♦ Las marcas no son interpretadas
 - ♦ Las macros no son interpretadas
 - ♦ Si se activa el fin del archivo (EOF), la zona verbatim acaba

Línea de Separación , Línea Gruesa

- **Descripción:** Identifica una línea de separación o una línea gruesa.
- **Propiedades:** !Multiline, FreeSpaces, !Align, !Nesting
- **Contenido:** --
- **Sintaxis:**
 - ◆ La línea de separación puede estar compuesta por guiones "-" o de caracteres de subrayado "_"
 - ◆ La línea gruesa está compuesta por signos igual "="
 - ◆ Deben usarse, al menos, 20 signos "-", "_" o "="
 - ◆ Tanto al principio como al final de la línea pueden colocarse espacios
 - ◆ Cualquier otro carácter colocado sobre la línea invalida la marca
- **Detalles:**
 - ◆ Si el archivo final no admite línea de separación, ésta es reemplazada por un comentario
 - ◆ La línea gruesa puede tener distintos aspectos en ciertos tipos de archivos finales:
 - ◇ Una línea de separación más gruesa
 - ◇ Una pausa, en los formatos de presentación del tipo MagicPoint
 - ◇ Un salto de página, en formatos con paginación, del tipo LaTeX

Enlaces, Enlaces nombrados

- **Descripción:** Identifica un enlace remoto (internet) o local
- **Propiedades:** !Multiline, !FreeSpaces, !Align, !Nesting
- **Contenido:** Macros, Raw, Imágenes
- **Sintaxis:**
 - ◆ Cualquier dirección internet válida (URL, Ftp, news, e-mail) es detectada y convertida automáticamente
 - ◆ La indicación del protocolo (http, https, ftp) es opcional : `www.comoesto.net`
 - ◆ Puede utilizarse un nombre para un enlace: `[clicar aqui www.url.net]`
 - ◆ Una imagen puede apuntar a un enlace: `[imagen.jpg www.url.net]`
 - ◆ Pueden usarse macros en las direcciones de los enlaces: `[ver source%infile]`
 - ◆ Pueden utilizarse macros en los nombres de enlaces: `[mirror de %%outfile www.debian.org]`
 - ◆ El enlace debe ser especificado en una sola línea, sin saltos de línea
- **Detalles:**
 - ◆ Si el archivo final no soporta enlaces, éstos serán subrayados.

Cita

- **Descripción:** Identifica una línea de cita (indentada).
- **Propiedades:** Multiline, !FreeSpaces, !Align, Nesting
- **Contenido:** Macros, Modificadores de Fuentes, Cita, Raw, Barras, Enlaces, Imagen, Comentarios
- **Sintaxis:**
 - ◆ Una línea que comienza por una tabulación (TAB)
 - ◆ A más TABs, mayor "profundidad"
 - ◆ No se permiten listas ni tablas
- **Detalles:**
 - ◆ Si el final del archivo fuente es activado (EOF) , se cierra la cita abierta
 - ◆ No hay límites en los niveles de cita, pero algunos tipos de archivos destino no soportan determinados niveles de cita. En estos casos, las sub-citas no soportadas son transformadas al nivel superior

Listas, Listas Numeradas, Listas de Definición

- **Descripción:** Identifica el comienzo de una lista de elementos
- **Propiedades:** Multiline, !FreeSpaces, !Align, Nesting
- **Contenido:** Macros, Modificadores de Fuentes, Listas, Tabla, Raw, Barras, Enlaces, Imágen, Comentario
- **Sintaxis:**
 - ◆ Una línea que comienza por la siguiente secuencia: guión "-" /dos puntos ":" / seguida de, exactamente, un espacio
 - ◆ El primer carácter de la lista no puede ser un espacio (excepción: listas de definición)
 - ◆ Espacios (no TAB) al principio de la línea definen la profundidad (niveles) de las sub-listas (anidamiento)
 - ◆ Las sub-listas se acaban con un elemento de menor profundidad o con un elemento vacío
 - ◆ Todas las listas abiertas se cierran con dos líneas en blanco consecutivas
- **Detalles:**
 - ◆ Si el final del archivo fuente es activado (EOF) se cierran todas las listas abiertas
 - ◆ Las listas pueden ser mixtas, por ejemplo, una lista de definición en una lista numerada
 - ◆ Algunos tipos de archivos finales no admiten la imbricación de listas, los elementos de las sub-listas son transformados al nivel superior..
 - ◆ No hay restricción en la profundidad de las listas, pero ciertos tipos de archivos destino tienen restricciones al respecto. En estos casos, las sub-listas de nivel inferior son ascendidas

Imágen

- **Descripción:** Identifica una imagen
- **Propiedades:** !Multiline, !FreeSpaces, Align, !Nesting
- **Contenido:** Macros
- **Sintaxis:**
 - ◆ Un archivo de imagen encerrado entre corchetes: [imagen.jpg]
 - ◆ El archivo de imagen debe acabar con una extensión del tipo PNG, JPG, GIF,.....(en mayúsculas o minúsculas, indiferentemente)
 - ◆ Se admiten símbolos en el nombre del archivo de imagen: [imagen~!tres.jpg]
 - ◆ Se admiten macros en el nombre del archivo de imagen:
[report-%%date(%Y-%m-%d).png]
 - ◆ No se admiten espacios en el nombre del archivo de imagen: [como esta.jpg]
 - ◆ No se admiten espacios entre el corchete y el nombre del archivo de imagen: [likethis.jpg] es incorrecto
- **Detalles:**
 - ◆ Si el archivo destino no admite imágenes, el nombre del archivo de la imagen se presenta entre (paréntesis)
 - ◆ La posición de las marcas indica el alineamiento de la imagen:
 - ◇ [IZQUIERDA.jpg] blablabla
 - ◇ blablabla [CENTRO.jpg] blablabla
 - ◇ blablabla [DERECHA.jpg]

Tabla

- **Descripción:** Delimita una fila de una tabla, de cualquier número de columnas.
- **Propiedades:** Multiline, FreeSpaces, Align, !Nesting
- **Contenido:** Macros, Modificadores de Fuentes, Raw, Enlaces, Comentario.
- **Sintaxis:**

- ◆ Una línea vertical "|" identifica una fila de una tabla.
- ◆ Dos líneas verticales "||" al principio de una línea identifica la fila de cabecera de una tabla
- ◆ Los espacios delante de la primera línea vertical personalizan el centrado de la tabla
- ◆ Los campos están separados por la cadena espacio/barra vertical/espacio " | "
- ◆ Una barra vertical "|" al final de la primera fila de una tabla hace visibles los bordes
- ◆ La barra-vertical al final de otras filas de la tabla es ignorada (sólo tiene efectos "cosméticos")
- ◆ Cerrando una celda con más de una barra vertical determina la expansión de columna (column span) : "||" para dos columnas, "|||" para tres columnas, etc....
- ◆ Los espacios(no TAB) en el interior de las celdas determinan su alineamiento.
- ◆ Ejemplo: | fila | de | tabla | con | seis | columnas |
- **Detalles:**
 - ◆ Cada fila de tabla debe estar en una sólo línea del archivo fuente, sin saltos de línea
 - ◆ Los archivos destino con alineamiento por columnas (como sgml y LaTeX) usan el alineamiento de la primera fila como referencia para las demás.
 - ◆ Toda línea "no-tabla" cierra la tabla abierta, con excepción de las líneas de comentario
 - ◆ El número de celdas es flexible.Cada fila puede contener un número distinto de celdas
 - ◆ No hay un modo de especificar la anchura de las columnas.
 - ◆ Si el archivo destino no admite tablas, las filas de la tabla son consideradas como una zona verbatim

Raw,Líneas y Zonas Raw (tal-cual)

- **Descripción:**Utilizado para proteger texto del archivo fuente, de manera que las marcas y macros que contenga no sean tratados
- **Propiedades:**!Multiline, !FreeSpaces, !Align, !Nesting
- **Contenido:** --
- **Sintaxis:Raw:**
 - ◆ Dos dobles-comillas alrededor del texto: " "como ésto" "
 - ◆ Sin espacios entre las marcas y el texto
- **Sintaxis:Línea Raw:**
 - ◆ Una línea que comienza con 3 dobles-comillas consecutivas:" " " línea
 - ◆ La marca debe estar en la primera columna de la línea, sin espacios delante
 - ◆ Utilizar un espacio después de la marca para separar la del texto
- **Sintaxis:Zona Raw:**
 - ◆ Una línea que comienza con 3 dobles-comillas consecutivas, seguidas de la línea de texto, seguida de otras 3 dobles-comillas consecutivas.
 - ◆ Sin espacios antes o después de las marcas
- **Detalles:**
 - ◆ Las marcas no son interpretadas
 - ◆ Las macros no son interpretadas
 - ◆ Si el final del archivo fuente es activado (EOF) se cierra la zona raw abierta

Parte VI – Dominando las Macros

Las macros son palabras-clave específicas que son utilizadas por el programa, en el momento de la conversión, para insertar informaciones dinámicas tales como la fecha e informaciones referentes al documento fuente o al documento final.

Una macro está constituida por los caracteres %% seguidos de su nombre, por ejemplo, %%date. Algunas macros aceptan una cadena de formato entre paréntesis, justo después del nombre de la macro, como, por ejemplo: %%date(%Y-%m-%d). Ésta cadena de formato incluye texto clásico junto con directivas, identificadas por %, seguidas de un carácter de identificación. Si no se especifica ninguna cadena de formato, se utilizará el formato por defecto.

Nombre Macro	Se transforma en...	Formato por defecto
%%date	Fecha actual	%Y%m%d
%%mtime	Fecha de modificación del archivo fuente	%Y%m%d
%%infile	Path al archivo fuente	%f
%%outfile	Path al archivo final	%f
%%toc	Índice de contenidos del documento (TOC)	–

Reglas generales:

- El nombre de la macros es insensible a la caja: %%date y %%DaTe son idénticas
- Las macros son válidas en las áreas del Encabezamiento y del Cuerpo del documento, excepto %%toc, que sólo es válida en el área del Cuerpo.
- Una macro que se encuentre en la zona de Configuración se ejecutará en la zona del Cuerpo.
- Una macro puede situarse en cualquier parte de una línea.
- Una línea puede contener varias macros. Excepción: %%toc, que debe estar sólo en la línea
- Una macro puede utilizarse en los enlaces y en las marcas de imagen (excepto %%toc)
- Las macros no son procesadas en los títulos ni en las zonas Verbatim y Raw

Ejemplo (el texto en negrita ha sido generado por macros):

Esto es la guía del Usuario de Txt2tags, convertida a **html** por txt2tags a partir del archivo fuente **userguide-es.t2t**. La conversión se realizó en fecha **2005-09-28 14:20:36**, los últimos cambios en el documento fuente fueron realizados el **2005-08-19 11:15:55**. Tanto el archivo fuente como el final se encuentran en el directorio **userguide-es**

%%date

Llamada sin ninguna directiva especial, %%date nos devuelve la fecha actual con el formato ISO yyyyymmdd. Puede cambiarse el formato usando la sintaxis %%date(formato)

Esta macro acepta un amplio conjunto de directivas. Puede encontrarse un listado completo de estas directivas en la web de [Phyton](#). Éstas son las directivas usadas con mayor frecuencia:

Directiva	Descripción
%a	Nombre abreviado del día de la semana.
%A	Nombre completo del día de la semana.
%b	Nombre abreviado del mes.

%B	Nombre completo del mes.
%c	Representacion del día y hora, con formato apropiado.
%d	Día del mes en formato decimal [01,31].
%H	Hora (modo 24–horas) en formato decimal [00,23].
%I	Hora (modo 12–horas) en formato decimal [01,12].
%m	Mes en formato decimal [01,12].
%M	Minutos en formato decimal [00,59].
%p	Equivalente local tanto para AM como PM.
%S	Segundos en formato decimal[00,61]. (1)
%x	Representación apropiada de la fecha local.
%X	Representación apropiada de la hora local.
%y	Año, sin las cifras de la centuria, en formato decimal [00,99].
%Y	Año, con las cifras de la centuria, en formato decimal
%%	El carácter %

Ejemplos

Macro	-->	Resultados convertidos en 2005, Sep 28 at 14:20
%%date(Convertido en: %c)	-->	Convertido en: Wed Sep 28 14:20:36 2005
%%date(%Y-%m-%d)	-->	2005-09-28
%%date(%l:%M %p)	-->	02:20 PM
%%date(Hoy es %A, on %B.)	-->	Hoy es Wednesday, on September.

%%mtime

Esta macro nos devuelve la fecha de la última modificación del archivo fuente. Acepta las mismas directivas de formato que [%%date macro](#).

Ejemplo:

Esta Guía del Usuario fué editada por última vez el **Fri Aug 19 11:15:55 2005**. Esta fecha ha sido generada por `%%mtime(%c)`.

%%infile

Nos devuelve el camino (PATH) al archivo fuente en el sistema. Es útil para hacer cosas como "mira la fuente de este archivo" en enlaces de páginas HTML. Dar esta información es un detalle amable hacia los debutantes, que podrán utilizar vuestra fuente como ejemplo para sus propias páginas.

Esta macro acepta las siguientes directivas de formato:

%<char>	Descripción	output generado para esta Guia del Usuario
%f	Nombre del archivo	userguide-es.t2t
%F	Nombre del archivo (sin	userguide-es

	extension)	
%e	Extensión del archivo	t2t
%p	Path absoluto al archivo	/a/txt2tags/src/doc/Spanish/userguide-es/userguide-es.t2t
%d	Path al archivo (sólo directorios)	/a/txt2tags/src/doc/Spanish/userguide-es
%D	Path al archivo (sólo directorio padre)	userguide-es
%%	Carácter %	%

Ejemplos:

Fuente	-->	Resultado
El directorio padre de esta guía es %%infile(%D).	-->	El directorio padre de ésta Guía es userguide-es.
Usa la extensión %%infile(%e) .	-->	Usa la extensión t2t.
[Consulta la fuente %%infile]	-->	Consulta la fuente
Convertido a XHTML, Se llamará %%infile(%F).xhtml	-->	Convertid a XHTML, se llamará userguide-es.xhtml

Nota: La macro nos devuelve "-" si el archivo fuente es STDIN.

%%outfile

Nos devuelve el camino (PATH) al archivo convertido, en el sistema. Es práctico colocado en las áreas de Encabezamiento y el Cuerpo del archivo. Acepta las mismas directivas que la macro [%%infile](#)

Ejemplos:

Fuente	-->	Resultado
Estás leyendo el archivo %%outfile .	-->	Estás leyendo el archivo userguide-pdf.html .
txt2tags -t %%outfile(%e) -i %%infile -o %%outfile	-->	txt2tags -t html -i userguide-es.t2t -o userguide-pdf.html

Nota: La macro nos devuelve "-" si el archivo fuente es STDOUT

%%toc

Esta macro nos devuelve el índice de materias del documento. Es útil para indicarnos donde lo queremos insertar. Puedes utilizar esta macro más de una vez para, por ejemplo, insertar el índice de materias al inicio y al final del documento . Esta Guía esta utilizando %%toc para insertar el índice de materias

A diferencia de otras macros, no acepta directivas de formato y se rige por unas reglas especiales:

- Sólo es válida si se coloca en el **Area del Cuerpo** del documento
- Debe encontrarse sólo en la línea (se permiten espacios antes y después de la macros)
- Debe utilizarse con la opción en línea de comandos --toc, si no, será ignorada.

- El posicionamiento y el formato por defecto del Índice de materias, se ignorarán cuando se encuentre una macro `%%toc`.

Parte VII – Dominando las Directivas de Configuración

Las Directivas de Configuración son directivas especiales, localizadas en el **Area de Configuración** del documento fuente, que afectan al comportamiento de la conversión.

Las Directivas de Configuración son todas opcionales. Cualquier usuario medio puede generar documentos txt2tags sin siquiera conocer la existencia de dichas Directivas, pero, como ya se ha dicho anteriormente, el usuario que las haya "experimentado" difícilmente podrá prescindir de las ventajas y el control que le proporcionan.

Las Directivas de Configuración se escriben como *líneas de comentario especiales*, precedidas de la marca ("!") que las diferencia de los comentarios clásicos. La sintaxis es tan simple como la de la asignación de un valor a una variable:

`%! palabra-clave : valor`

Detalles de la sintaxis:

- No debe haber espacio entre el carácter de comentario y el signo de exclamación ("%!")
- Los espacios entre la clave y el separador (":") son opcionales
- La clave y el valor son insensibles a la caja (no distinguen entre mayúsculas y minúsculas)

Reglas:

- Sólo son válidas en el **Area de Configuración**. Si se emplazan en otra área del documento serán tratadas como comentarios
- Si la misma directiva aparece varias veces en el **Area de Configuración**, sólo se tomará en consideración la última. Excepción: options, preproc y postproc, que son acumulativas
- Una directiva con una clave no válida será considerada un comentario

- Las directivas tienen prioridad sobre el archivo RC pero no sobre las opciones en línea de comandos

%!Target

Usando la configuración target se define un formato para el documento destino:

```
%!target: html
```

Así, al usuario le bastará con ordenar

```
$ txt2tags file.t2t
```

Y la conversión se realizará al formato especificado en target

Target no permite una especificación opcional de destino. Así, no tendrá sentido: `%!target(tex):html`.

%!Options

Tener que escribir largas líneas de comando cada vez que sea necesario convertir un documento, además de cansado puede ser fuente de numerosos errores. La configuración `%!options` permite que el usuario guarde, en el propio documento origen, todas las opciones de configuración que necesitará. Esto es garantía, además, de que el documento será siempre convertido de la misma

forma, con las mismas opciones.

Las opciones deben escribirse como se haría en la línea de comandos, sin errores sintácticos. Pero no debe incluirse la llamada al programa "txt2tags" ni el nombre del archivo origen al final de la línea. Si en línea de comandos se utiliza la siguiente orden:

```
$ txt2tags -t html --toc --toc-level 2 --enum-title file.t2t
```

En la configuración `%!options` deberá escribirse:

```
%!target: html
%!options(html): --toc --toc-level 2 --enum-title
```

La línea de comandos, para la conversión del documento que incluya la configuración `%!options`, pasa a ser "txt2tags file.t2t", y la conversión puede ser ejecutada dentro nuestro editor de texto preferido mientras el contenido está siendo editado. En Vi, el comando es:

```
:!txt2tags %
```

%!Encoding

La configuración de `%!encoding` es casi imprescindible para aquellos autores que escriben en idiomas que contemplan letras acentuadas y otros detalles específicos, de forma que el Conjunto de Caracteres del documento final pueda ser personalizado. Los valores válidos para la configuración de `%!encoding` son los mismos que para documentos HTML, (iso-8859-1, UTF-8, koi8-r, etc.). En la siguiente dirección puede consultarse la lista de [códigos](#)

El documento final LaTeX utiliza alias para la codificación . Esto no representa ningún problema para el usuario, ya que txt2tags los traduce internamente. Algunos ejemplos de esta traducción son los que a continuación se relacionan:

txt2tags/HTML	>	LaTeX
windows-1250	>>>	cp1250
windows-1252	>>>	cp1252
ibm850	>>>	cp850
ibm852	>>>	cp852
iso-8859-1	>>>	latin1
iso-8859-2	>>>	latin2
koi8-r	>>>	koi8-r

Si el valor no es reconocido por txt2tags, será pasado sin codificar, permitiendo al usuario especificar codificaciones personalizadas.

%!PreProc

Es un filtro con la función "busca y cambia", que es aplicada en el documento original, después de que la línea haya sido leída y antes de cualquier tratamiento por txt2tags.

Es útil para definir algunas abreviaciones para texto que sea de uso frecuente, como:

```
%!preproc JJS "John J. Smith"
%!preproc RELEASE_DATE "2003-05-01"
```

```
%!preproc BULLET "[images/tiny/bullet_blue.png]"
```

Si el usuario escribe una línea como ésta :

```
¡Hola! Me llamo Antoni. Hoy es RELEASE_DATE.
```

txt2tags la presentará como :

```
¡Hola! Me llamo Antoni. Hoy es 2003-05-01.
```

Este filtro es un componente que trabaja entre el autor del documento y la conversión de txt2tags. Es una especie de "primera conversión" antes de la conversión definitiva. Este componente funciona exactamente como un filtro Sed/Perl externo, que se invocan de la siguiente forma:

```
$ cat ARCHIVO.t2t | preproc-script.sh | txt2tags -
```

El tratamiento ejecutado por txt2tags comenzará una vez todas las sustituciones PreProc hayan sido realizadas.

%!PostProc

Es un filtro con la función "busca y sustituye", que es aplicada al archivo resultante, una vez realizadas todas las conversiones por txt2tags.

Es útil para definir algunos refinamientos en el documento generado, cambiar algunas marcas, añadir texto extra, etc. Algunos breves ejemplos:

```
%!postproc(html) : '<BODY.*?>' '<BODY BGCOLOR="green">'
%!postproc(tex)  : "\\newpage" ""
```

Estos filtros cambian el color de fondo de la página HTML y eliminan los saltos de página en archivos LaTeX generados.

Las reglas de PostProc son las mismas que las de los filtros externos Sed/Perl, que se invocan de la siguiente forma:

```
$ txt2tags -t html -o- file.t2t | postproc-script.sh > file.html
```

Antes de que apareciera esta funcionalidad, era bastante común tener pequeños scripts para "ajustar" los resultados generados por txt2tags. Dichos scripts eran, de hecho, varios comandos sed (o similares), para hacer algo parecido a "sustituye esto por aquello". Actualmente las órdenes para estas sustituciones pueden ser guardadas junto con el texto del documento, y también aprovechan de la poderosa máquina de Expresiones regulares de Python.

%!Style

- Esta opción es útil en la generación de archivos HTML y XHTML, definiendo el nombre del archivo CSS que será usado por el documento generado.
- En el formato LaTeX, esa opción se usa para cargar módulos /usepackage.
- El mismo resultado se obtiene con la opción en línea de comandos `--style`
- La opción `--style` en línea de comandos tiene preferencia sobre la configuración de `%!style`.

Definiendo una Configuración para un Tipo de Archivo Destino Específico

Desde la versión 1.6 de `txt2tags`, todas las Directivas de Configuración, excepto `%!target`, pueden ser especificadas para un tipo de archivo destino específico, usando la sintaxis :

`%!clave(destino):valor`. De esta forma, el usuario puede definir diferentes configuraciones para diferentes tipos de archivos generados.

Éste es un aspecto especialmente útil en los filtros `pre/postproc`, aunque puede ser utilizado en todas las directrices. Por ejemplo, configurar diferentes estilos para HTML y LaTeX:

```
%!style(html): fancy.css
%!style(tex) : amssymb
```

Para ajustar las opciones en el documento convertido, algo como ésto puede ser útil:

```
%!target: sgml
%!options(sgml): --toc
%!options(html): --style foo.css
%!options(txt) : --toc-only --toc-level 2
```

De esta forma, el tipo de destino patrón es SGML y generará un índice de materias (TOC). Si el usuario ejecuta:

```
$ txt2tags -t html file.t2t
```

El tipo de destino será HTML y solamente serán usadas las opciones de `%!options(html)`. De esta forma se utilizará la opción `---style` y en el archivo HTML no se generará un índice de materias.

Detalles de los filtros PreProc y PostProc

- Los filtros contienen una función de "buscar y sustituir".
- No son del tipo "utiliza la última encontrada", ya que son acumulativos. Podemos definir tantos filtros como deseemos. Serán aplicados todos, en el orden en que hayan sido definidos.
- Para los filtros, el orden de definición marca su aplicación y serán aplicadas tanto las reglas genéricas como las específicas al tipo-destino. En el siguiente ejemplo, ambos filtros son aplicados en el documento HTML final:

```
%!postproc      :   this   that
%!postproc(html):   that   other
```

- Los filtros deben recibir exactamente dos argumentos.
- Escapes especiales como `\n` y `\t` también son interpretados.
- Para eliminar texto, sustitúyelo por una cadena vacía

```
%!postproc: "texto no deseado" ""
```

- Para evitar problemas, procura siempre indicar explícitamente el destino al que se aplicará el filtro cuando se esté usando PostProc para cambiar marcas: `%!PostProc(destino):`
`<esto>` `<aquello>`
- PreProc es aplicado después de que una línea haya sido leída. PostProc es aplicado después de que todo el tratamiento se haya realizado. Funcionan exactamente como en:

```
$ cat file.t2t | preproc.sh | txt2tags | postproc.sh
```

- La primera parte de un filtro (la cadena "busca"), no es leída como una cadena sino como una expresión regular. Si no sabes lo que es ésto, no te preocupes, casi seguro que nunca tendrás necesidad de ello. Pero recuerda que debes "escapar" los caracteres para el

usuario. "Escapar" significa preceder el carácter con el signo "\". Cómo en la siguiente lista:

```
\* \+ \. \^ \$ \? \( \) \{ \[ \| \\
```

- Las Expresiones Regulares de Python pueden ser utilizadas. Son similares a las Regex de Perl. Por ejemplo: cambia todas las marcas "B" para "STRONG" en HTML:

```
%!postproc(html): '(</?)B>' '\1STRONG>'
```

- Los argumentos de los filtros pueden ser pasados de tres formas:
 1. Una palabra simple, sin apóstrofes ni comillas, como FOO (sin espacios)
 2. Una cadena con dobles comillas, como en "FOO"
 3. Una cadena ,con comillas o apóstrofes, como en 'FOO'
- Si la expresión origen tiene apóstrofes, protéjelos con comillas y vice-versa. Algunos ejemplos:

```
%!postproc: ORIGEN CAMBIO
%!postproc: "ORIGEN" "CAMBIO"
%!postproc: 'ORIGEN' 'CAMBIO'
%!postproc: ORIGEN "CAMBIO"
%!postproc: "ORIGEN" 'CAMBIO'
```


Parte VIII – Mágia Negra

Este no es un capítulo recomendado para novatos. Aquí se va a mostrar como hacer cosas "extrañas" con los filtros de txt2tags, "abusando" de patrones complejos y de Expresiones Regulares.

¡Atencion! Los procedimientos detallados a continuación pueden acarrear daños. Algún texto puede perderse en el procedimiento de conversión, no apareciendo en el documento generado. Usa estos procedimientos sólo si precisas verdaderamente de ellos y/o si sabes exactamente lo que estas haciendo

Los filtros son un procedimiento poderoso, pero pueden ser peligrosos

Los filtros mal usados generan resultados inesperados

Insertando Múltiples Líneas con %!PostProc (como reglas CSS)

En los filtros, el patrón de sustitución puede incluir múltiples líneas utilizando el carácter de salto de línea \n

Esta funcionalidad puede ser útil para incluir reglas CSS en un archivo HTML generado, sin necesidad de crear un archivo separado. Este es el caso de esta Guia del Usuario, que utiliza los siguientes filtros:

```
%!postproc: <HEAD>      '<HEAD>\n<STYLE TYPE="text/css">\n</STYLE>'  
%!postproc: (</STYLE>) 'body      { margin:3em          ;} \n\n1'  
%!postproc: (</STYLE>) 'a         { text-decoration:none ;} \n\n1'  
%!postproc: (</STYLE>) 'pre,code { background-color:#ffffcc ;} \n\n1'  
%!postproc: (</STYLE>) 'th        { background-color:yellow ;} \n\n1'
```

Todos los filtros están subordinados al primero, para reemplazar una cadena que ha insertado. De esta forma, un simple "<HEAD>" se transforma en:

```
<HEAD>  
<STYLE TYPE="text/css">  
body      { margin:3em          ;}  
a         { text-decoration:none ;}  
pre,code  { background-color:#ffffcc ;}  
th        { background-color:yellow ;}  
</STYLE>
```

Creando Contenido "Específico al Tipo Destino" con %!PreProc

Algunas veces es necesario insertar algún texto solo para un tipo de destino específico y no para otros. Este tipo de comportamiento "extraño" puede ser llevado a cabo con PreProc.

La idea es insertar este texto extra en el archivo original como comentario, pero marcarlo de forma que un filtro específico para aquel tipo de destino lo "descomente"

Por ejemplo, si debe insertarse un párrafo extra solamente en el tipo de destino HTML, coloca el texto en un comentario especial como este:

```
%html% Esta página HTML fué generada por [txt2tags http://txt2tags.sf.net].  
%html% Vea el archivo fuente TXT [aquí source.t2t].
```

Como todas las líneas comienzan con %, son líneas de comentario y serán ignoradas. Pero si

añadimos este filtro especial:

```
%preproc(html): '^%html%'
```

El signo % inicial es eliminado, "activando" las líneas que lo contenían, que dejarán de ser comentarios para pasar a integrar el texto del documento final. Como hemos configurado un destino específico, HTML, el filtro sólo se aplicará a él.

Cambiando Marcas txt2tags con %!PreProc

Si el usuario es un gurú de Expresiones Regulares, podría personalizar la sintaxis del documento, cambiando las marcas txt2tags patrón por algo que le fuere mas confortable.

Por ejemplo, una TAB es una marca de cita. Si al usuario no le gusta eso, puede definir una nueva marca para el "texto citado". Consideremos que escoja colocar ">>>". Precisaré de este filtro simple:

```
%!PreProc: '>>>' '\t'
```

Y en el documento original, el "texto citado" aparecera como:

```
>>> Este es un texto citado
>>> El usuario definio esta marca exquisita
>>> Pero seran convertidas a TABs por PreProc
```

Antes que el analisis del programa comience, las ">>>" seran transformadas en TABs y txt2tags reconocera la marca de cita.

¡Atencion! Las Reglas PreProc extremas pueden ,eventualmente, cambiar toda la sintaxis de marcado y asi mismo, generar conflictos entre marcas. Ten mucho cuidado en la utilizacion de estas reglas.

Parte IX – História de Txt2tags

En julio de 2001 se lanzó al público la primera versión de txt2tags (v0.1). Pero sus orígenes se remontan a más de un año antes de aquel día.

Este capítulo ilustra, en pocas palabras, el desarrollo de esta herramienta, desde que la primera línea fue escrita hasta la actual versión.

Enero de 1999: Pre-História

"Mi primera tentativa de hacer una herramienta de conversión comenzó en 1999, como un simple script Bourne Shell que convertía texto con marcas en una página HTML. Como otra herramienta txt2html. Todo el mundo debe haber hecho ya una de estas herramientas.....En resumen, reconocía marcas simples, como **bold**, */italic/*, *_subrayado_*, y escapaba los caracteres especiales `< & >` de HTML. Nada impresionante, pero era joven;)"

Júlio de 1999: Todavía Pre-História

"Pasaron algunos meses, y un gran "hype" entorno a Sgml llegó a la compañía en la que estaba trabajando (Conectiva). Txt2tags se transformó en un script txt2sgml. Estaba realmente tentado a aprender sobre SED* en aquel tiempo, de forma que txt2tags en aquella época era un script Bourne Shell con mucho código SED"

* **SED**: UNIX Stream EDitor – una herramienta de edición automática de textos

Esta versión mejorada, con soporte Sgml, ofrecía soporte a más estructuras como listas y texto preformateado. En el siguiente archivo ejemplo, pueden verse los orígenes de las marcas txt2tags:

```
* Esta es una línea en negrita (¿NEGRITA orientada a línea? bien...)

--
- lista no numerada muy semejante a lista de txt2tags
- pero con éstos -- para iniciar y terminar la lista
--

=====
El texto preformateado era delimitado por el patrón =-- .
Los otros ----- eran cosméticos.
=====
```

Nada impresionante todavía, pero el gran paso está al caer...

Agosto de 2000: No más Pre-História

"Pasó un año y, durante este tiempo, estuve realmente encantado con SED. El script de shell txt2sgml.sh fue reescrito convirtiéndose en 350 líneas de puro script SED. Fueron añadidas algunas funcionalidades como subsecciones, reconocimiento de URL y sublistas. Durante al menos un año, usé y probé muchísimo dicho script

Un sencillo ejemplo de archivo txt2sgml.sed :

```
* Hey, here are the first 3 magic lines
* The document title / author / date
* But they required those asterisks at the beginning
```

MAIN TITLE

Titles were made by uppercase-only lines. Subtitles were identified by leading spaces. Each space denoted a new sublevel. The beautifiers: `*bold*`, `**strong**`, `"italic"` and ``typewriter``.

```
- lists
+ sublists
  = and subsublists (by identifier, not indentation)
```

Two blank lines to close lists. Links as `www.example.com` and e-mails were recognized automagically by regular expressions. And there was a strange image mark:
`%image: path/to/image.jpg`

Mayo 2001: Pythonización e Idea Multi Formato de Destino

"Había comenzado a escribir mi libro, [Expresiones Regulares](#) y usado el formato de marcado `txt2sgml.sed`. De esta forma pude convertirlo a `sgml` (luego a HTML utilizando una herramienta `sgml2html`) y chequear rápidamente en el navegador cómo había quedado el libro. Pero cómo "rápido" y "sgml2html" no casan, modifiqué el script `SED` de `txt2html.sed` generando directamente HTML. [...]. Mi editor usaba Adobe PageMaker para el formateo de libros, lo que era un problema para un chico linux como yo. Pero me alegré cuando me enteré que PageMaker utilizaba un sistema de marcas semejante al de HTML, entonces, comencé a cambiar mi script en una herramienta `txt2pagemaker.sed`. Había acabado por tener tres scripts `SED` similares, convirtiendo mis textos a `sgml`, HTML y PageMaker. Se crearon algunos scripts más para extraer el índice de contenidos (TOC) del libro y para realizar post-formateos. A mitad redacción del libro, tuve la idea de unirlos en una sola herramienta y elegir Python como lenguaje. Nació `TXT2TAGS`"

Júlio de 2001: séries 0.x: Debut de txt2tags (Lanzamiento Mundial)

"El lanzamiento del libro impreso (31 de julio) y de la primera versión (0.1) de `txt2tags` (26 de julio) fue asunto de días. Uno dependía de otro, fueron desarrollados juntos. Para ésta entrega, junto con `sgml`, HTML y PageMaker, se implementaron otros formatos de archivo destino: MoinMoin, MagicPoint y texto plano. En menos de un año se realizaron más versiones de la série 0.x y el programa comenzó a crecer: nuevo formato UNIX Man page, macro `%%date`, soporte para tablas, interfaz Web, alineamiento de imagen y generación de índices de contenido (TOC). En la versión 0.2 se añadió un script de Shell para el manejo de operaciones de archivo y opciones en línea de comando. Lo añadí por que yo era un mago del shell y un novato de Python. No es hasta la versión 0.9 cuando `txt2tags` se transforma en 100% código Python."

Septiembre de 2002: séries 1.x : Crecimiento

"La idea `txt2tags` resultó ser una buena idea. Decidí tomármela en serio. La siguiente etapa fue la de expandir el programa, dándolo a conocer: ¡documentación!. Fue lanzado el site del programa, se configuraron listas de correo (en inglés y portugués) y nació la Guía del Usuario. La base de usuarios creció y llegaron numerosas contribuciones. Sí!, pensé, está funcionando. Las nuevas funcionalidades añadidas a las versiones de la série 1.x incluyen: Interfaz Gráfica (GUI), compatibilidad Windows

& Mac, formato de destino LaTeX, %!style , el comando include y los potentes filtros de Pre y Post proceso. "

Júlio de 2004: séries 2.x : Maduración

"El crecimiento es duro y extraño. ¿Recuerdas que dije que era un novato de Python?. Ahora estoy mejor, pero los viejos fallos acumulados, y el reescribir la mayor parte del código lo hicieron inevitable. Esto rompió la compatibilidad con algunas marcas, por lo que se creó un script actualizado. Se lanzó la versión 2.0. Con ella llegó el formato XHTML, la validación de código W3C, i18n y el archivo RC. Un equipo de traductores hacen que el programa y su documentación sea traducida a varios idiomas. Se añadió el formato de destino Lout y las nuevas macros %%mtime, %%infile, %%outfile, y %%toc. Continúa.....

The End.



