

# Sinopsis del formato markdown para pandoc<sup>\*,\*\*</sup>

Joaquín Ataz López

Febrero de 2012

## Índice

<b>1. Markdown y pandoc</b>	<b>2</b>
1.1. Qué es markdown . . . . .	2
1.2. Qué es pandoc . . . . .	3
1.3. Sintaxis de las llamadas a pandoc . . . . .	5
<b>2. FORMATO MARKDOWN PARA PANDOC</b>	<b>7</b>
2.1. Estructura del documento . . . . .	8
2.1.1. Secciones . . . . .	8
2.1.2. Identificadores de la sección para referencias cruzadas . .	9
2.2. Comentarios . . . . .	10
2.3. Formatos de carácter . . . . .	11
2.4. Formatos de párrafo . . . . .	12
2.4.1. Párrafos normales . . . . .	12
2.4.2. Listas . . . . .	12
2.4.3. Párrafos verbatim (literales) . . . . .	15
2.4.4. Blockquotes (citas) . . . . .	15
2.5. Notas a pie de página . . . . .	15
2.6. Tablas . . . . .	16

---

<sup>\*</sup>Este documento se distribuye con licencia [LGPL](#)

<sup>\*\*</sup>Para generar el fichero en formato .pdf este documento se este documento se procesa con pandoc: `pandoc --template=plantilla.tex -t latex --toc -N markdown.mkd | pdflatex`. El fichero fuente y la plantilla estan en: [markdown-umu](#)

2.6.1. Tablas simples . . . . .	16
2.6.2. Tablas multilínea . . . . .	18
2.6.3. Tablas cuadriculadas . . . . .	18
2.7. Otros elementos en el texto . . . . .	18
2.7.1. Enlaces: . . . . .	18
2.7.2. Imágenes: . . . . .	19
2.7.3. Bloques de título: . . . . .	20
2.7.4. Caracteres literales: . . . . .	21
2.7.5. Líneas horizontales: . . . . .	21
2.8. Resumen de las reglas de indentación . . . . .	21

## 1. Markdown y pandoc

### 1.1. Qué es markdown

Markdown es un lenguaje de marcas diseñado para conseguir la máxima legibilidad del documento fuente. Para entender lo anterior hay que aclarar estos dos conceptos: lenguajes de marcas y legibilidad.

**Lenguajes de marcas:** (o de marcado) consisten en un conjunto de convenciones en virtud de las cuales en un documento de texto puede incluirse, junto con el texto propiamente dicho que constituye su contenido, información relativa a la estructura o formateo del documento. Esta información se introduce mediante marcas o etiquetas que también son de texto. Ejemplos de estos lenguajes son latex, html, rtf, xml, etc.

Estos ficheros son siempre ficheros de texto plano y pueden ser leídos en su forma original con un editor de textos. Pero habitualmente su extensión no es txt, sino una extensión indicativa del lenguaje de marcas usado, como, por ejemplo, xml, html, tex, rst, rtf. . . En markdown se acostumbra a usar la extensión mkd, o md.

La extensión del fichero suele indicar al sistema operativo que estos ficheros deben abrirse por defecto, no con un editor de textos, sino con algún programa específico capaz de leer e interpretar las marcas del concreto lenguaje; de tal manera que el usuario normal, no llega a ver el documento tal y como fue escrito, sino que directamente se muestra el resultado de su procesamiento. Por ejemplo: un fichero html casi nunca es leído tal y como está escrito, sino a través de un navegador web (Internet Explorer, Firefox. . .) que analiza las marcas y las usa para formatear el texto, de manera que el usuario no ve las

marcas propiamente dichas, sino el texto ya formateado. Igual ocurre con los ficheros docx o rtf: haciendo doble click sobre ellos el sistema operativo abrirá la aplicación encargada de procesar estos ficheros (en sistemas windows, Microsoft-Word; en otros sistemas, normalmente Open Office).

**Legibilidad:** para las personas no acostumbradas a manejar lenguajes de marcas, los ficheros originales son difíciles de comprender. Están llenos de textos o etiquetas que hacen difícil leer el contenido del fichero para quien no conozca el lenguaje.

Pues bien: la legibilidad del lenguaje de marcas tiene que ver con la facilidad para leer el documento original sin haberlo procesado. Los lenguajes cuyas marcas son más sencillas, resultan mucho más legibles. En un lenguaje muy legible podemos leer el fichero original y extraer de él su información sin necesidad de procesarlo e incluso sin necesidad de comprender totalmente el lenguaje de marcas.

Markdown consigue su legibilidad haciendo que las marcas consistan exclusivamente en símbolos, caracteres especiales y niveles de sangrado. De este modo todas las “palabras” del documento forman parte de su información: leyendo el documento quizás no veamos cómo quedaría una vez formateado, pero podemos comprender perfectamente su contenido, sin etiquetas o textos extraños que distraigan nuestra atención.

Pero la principal utilidad de markdown está en lo sencillo que es convertir desde markdown hacia cualquier otro lenguaje de marcas. Para ello hay una aplicación fantástica llamada pandoc.

## 1.2. Qué es pandoc

Pandoc es un programa capaz de convertir ficheros desde un lenguaje de marcado a otro lenguaje de marcado distinto. Es decir: el programa lee un fichero escrito en un lenguaje de marcado, y, a partir de él, crea un fichero nuevo, en un lenguaje de marcado distinto. Los lenguajes de marcado que pandoc puede manejar son los siguientes:

[pos = H, center, botcap]ll A partir de cualquiera de estos lenguajes pandoc puede generar un fichero en cualquier otro de dichos lenguajes, y también, entre otros, en:

- ConText
- rtf
- DocBook XML

- Open Document XML
- odt
- TeXinfo
- MediaWiki markup
- man
- Microsoft Word docx
- epub

El lenguaje nativo para la entrada de pandoc es markdown; pero no un markdown puro y estricto, sino una versión ampliada del lenguaje. En este documento se explica esa versión, por lo que los ficheros markdown escritos con el formato que aquí se indica, podrían no funcionar correctamente si son tratados con aplicaciones distintas de pandoc.

Para quienes gustamos de escribir en texto plano, pero tenemos que relacionarnos con personas, colegas o editoriales que exigen documentos en formato Word (o similares) pandoc + markdown constituyen la solución ideal: Podemos escribir el documento en markdown y a partir de él generar un fichero docx u odt para enviar.

A mí me gusta escribir mis documentos en LaTeX, pues la calidad tipográfica que se obtiene es soberbia y, además, no me siento a gusto manejando un procesador de textos: trabajando directamente con texto plano tengo a mi disposición la potencia de las expresiones regulares (que no suelen existir en programas tipo Microsoft Word) así como la posibilidad de incluir comentarios en mis documentos. Pero las editoriales suelen exigirme que presente mis originales en doc (lo que, por otra parte, habla fatal de su cultura tipográfica). Esto era para mí un problema hasta que descubrí markdown + pandoc, porque no hay un buen sistema de conversión entre LaTeX y Microsoft Word<sup>1</sup>, y, como he dicho antes, no me gusta demasiado usar procesadores de texto<sup>2</sup>.

Pero markdown constituye una solución intermedia: puedo escribir el texto en markdown, y a partir de él generar versiones en LaTeX para almacenar yo mis documentos terminados, y versiones en Microsoft Word para enviar a la editorial.

---

<sup>1</sup>El mejor sistema de conversión que conozco es latex2rtf, pero su uso exige planificar un LaTeX bastante simple, en el que apenas se pueden usar paquetes de extensión, y luego el RTF resultante hay que reformatearlo a mano pues ciertos aspectos, como por ejemplo las notas a pie de página, no quedan en un formato satisfactorio.

<sup>2</sup>Sobre la diferencia entre “editores de texto” y “procesadores de texto” puede verse <http://webs.um.es/jal/43-edtextos.xml>. Allí explico, también, por qué prefiero trabajar con ficheros de texto en lugar de con ficheros binarios. En un sistema como linux, además, en donde existen numerosas utilidades para el tratamiento de ficheros de texto (cat, diff, grep, head, sed, sort, tac, tail, uniq, wc...) no usarlos como opción principal es un absoluto desperdicio.

Puedo, además, controlar casi completamente la conversión a Word mediante alguna de las opciones de pandoc.

Por ejemplo, si mi fichero original se denomina “origen.mkd” la orden para generar un documento latex a partir de él sería:

```
pandoc -s -o destino.tex origen.mkd
```

Ese documento podría ser formateado con LaTeX. Pero incluso puedo conseguirlo todo con una sola orden, aprovechando las tuberías de unix:

```
pandoc -s -t latex origen.mkd | pdflatex
```

Esto generaría un fichero llamado texput.pdf con la versión ya compilada en latex de mi fichero mkd.markdown.

En teoría lo mismo podría conseguir con `pandoc -s origen.mkd -o destino.pdf` pero por alguna razón, a mí no me funciona.

Igualmente, para enviar a la editorial una versión en Word de mi documento bastaría con:

```
pandoc -s origen.mkd -o destino.docx
```

aunque yo en la práctica prefiero:

```
pandoc -s --reference-odt=plantilla.odt origen.mkd -o destino.odt
```

Esto genera una versión en Open Office, fácilmente convertible a Word desde el propio Open Office, pero también susceptible de ser enviado directamente a la editorial, pues, aunque muchas editoriales no conozcan LaTeX si suelen conocer Open Office. La ventaja de este procedimiento frente al anterior está en que destino.odt tendrá exactamente las mismas características de formato que tuviera el fichero plantilla.odt, que previamente debemos haber creado y ajustado a nuestras preferencias, o a las indicaciones de la editorial.

### 1.3. Sintaxis de las llamadas a pandoc

Pandoc es una orden de consola. Su formato general es el siguiente:

```
pandoc [OPCIONES] FicheroAConvertir [OPCIONES]
```

Es decir: Las opciones se pueden colocar, indistintamente, antes o después del nombre del fichero a convertir; o, dicho de otro modo: pandoc considerará que el fichero a convertir es aquel nombre de la línea de comandos que no consiste en una opción.

Habitualmente pandoc detecta el formato original del fichero a convertir a partir de su extensión. Pero si la extensión no se correspondiera con el formato interno del fichero, o no fuera lo suficientemente explícita (por ejemplo: txt), pandoc asumirá que el formato es markdown.

Las opciones más importantes de pandoc (las que yo más uso) son:

- **-f**: Indica explícitamente el formato del fichero de origen. Si se introduce esta opción, pandoc ignorará el formato que cabría deducir de la extensión del fichero. Esta opción sólo hay que usarla, por lo tanto, cuando el formato real del fichero no se corresponda con su extensión, o cuando la extensión no represente correctamente el formato de nuestro fichero; circunstancias ambas que ocurren con muy poca frecuencia.
- **-o**: Nombre del fichero de salida. Si no se indica ninguno la salida se hará hacia la pantalla. Esta opción es obligatoria con algunos formatos de salida, en particular odt, docx, epub y pdf.
- **-t**: Formato de salida. Puede ser, entre otros<sup>3</sup>, beamer, context, docbook, docx, epub, html, html5, latex, man, markdown, mediawiki, odt, opendocument, rtf o texinfo,

Cuando se indica un nombre para el fichero de salida pandoc puede deducir el formato de salida a partir de dicho nombre, por lo que no habría por qué indicar explícitamente esta opción, la cual, por lo tanto, sólo se suele usar cuando no se use **-o**, o cuando la extensión del fichero de salida no sea lo suficientemente explícita respecto al formato al que deseamos convertir nuestro fichero original.

- **-s**: En los lenguajes de marcas es corriente que los documentos contengan un encabezado con metainformación necesaria para procesar el fichero. Por defecto pandoc se limita a convertir las marcas de un lenguaje a otro; pero usando la opción **-s** conseguiremos que se genere un documento completo a partir de las plantillas que, en un sistema linux ordinario, se almacenan en `/usr/share/versión-de-pandoc/templates`.

Modificando dichas plantillas podemos alterar la forma en la que se generan los documentos. Por ejemplo, para conseguir que en latex el guionado de las

---

<sup>3</sup>Además de los indicados en el texto se puede convertir también a formato nativo de hskell (native), texto plano sin marcas (plain) y otros formatos menos corrientes que yo, en general, no conozco, como asciidoc, dzslides, json, org, rst, s5, slidy, o textile.

palabras respete las reglas del español, deberíamos modificar el documento denominado “`latex.template`” para incluir en la plantilla la orden `\usepackage[spanish]{babel}`.

Pero, salvo para una modificación muy simple y que necesitaremos siempre, como la que se ha puesto de ejemplo, no se recomienda modificar directamente las plantillas originales. Es preferible, a partir de ellas, crear una plantilla modificada, e indicarle a pandoc que la use en lugar de la plantilla por defecto. Esto se hace mediante la opción `--template=fichero`, donde “fichero” sería el nombre de nuestra plantilla modificada para la ocasión.

Por otra parte, al usar la opción `--template` no es preciso usar la opción `-s` ya que ésta va implícita en aquella.

Por último diré que en mis pruebas se produce un error en la codificación de los ficheros docx cuando con ellos no se usa la opción `-s`; cosa que no entiendo porque no hay un template para docx.

- **`--reference-odt=fichero.odt`**: Cuando creamos un docx no controlamos muy bien el formato de salida. Por eso es siempre preferible generar un odt (a partir del cual podemos, si queremos, generar un docx con open-office). Podemos crear un documento odt en el que hayamos definido el formato general de párrafos, encabezados, notas a pie... y luego indicar a pandoc, mediante esta opción, que use dicho documento como plantilla para el nuevo documento a crear.
- **`--toc`**: En determinados formatos de salida (latex, html), genera automáticamente un índice a partir de la estructura del documento.
- **`-N`**: Hace que las secciones y subsecciones se numeren. Si no se incluye esta opción las secciones de nuestro documento no serán numeradas automáticamente.

Obsérvese, por otra parte, que cuando la opción exige un nombre, este se separa de la opción por un espacio en blanco, salvo en `--reference-odt` y en `--template` en las que el nombre del fichero se separa de la opción por el signo “=”.

## 2. FORMATO MARKDOWN PARA PANDOC

Pandoc introduce algunos cambios en el estándar markdown, dirigidos a hacer a markdown más versátil y potente. Si nuestro documento fuente usa markdown puro, podemos indicárselo a pandoc mediante la opción “`--strict`”; pero yo no aconsejo usar markdown puro: las extensiones que pandoc introduce en markdown lo hacen mucho más potente.

A continuación explicaré la sintaxis fundamental de markdown, tal y como lo maneja pandoc. Como este lenguaje es una mezcla de markdown y pandoc casi podríamos llamarlo “panmarkdown”. No lo hago así por no liar más las cosas.

Téngase en cuenta que el presente documento no es completo. Se han omitido varios aspectos de markdown + pandoc entre los que podemos destacar:

- Código HTML y TEX bruto.
- Uso de macros.
- Citas bibliográficas.
- Generación de diapositivas con pandoc.

Toda esta información se puede encontrar en <http://johnmacfarlane.net/pandoc/README.html>

## 2.1. Estructura del documento

### 2.1.1. Secciones

Un documento bien estructurado consta de secciones, las cuales a su vez pueden contener otras secciones, y así sucesivamente. Según el tipo de documento cada nivel de las secciones recibe un nombre distinto y así, por ejemplo, en un libro podemos distinguir entre partes, capítulos, subcapítulos...

En markdown a todos estos encabezados se les llama genéricamente secciones, con independencia de su nivel. Se admiten hasta seis niveles de profundidad, y la forma de indicar que empieza una sección es muy simple, basta con introducir una línea que, estando precedida de una línea en blanco (salvo que sea la primera línea del documento), empiece con uno o más caracteres “#”. Por ejemplo:

```
# Esto es un título de ejemplo
```

Respecto de la indicación de las secciones por este procedimiento debe tenerse en cuenta que:

- El número de caracteres “#” indica el nivel o profundidad de la sección, es decir: “##” empezará una sección de nivel 2 y “####” una sección de nivel 4.
- Entre los distintos “#” no puede haber espacios en blanco.



- El texto que vaya inmediatamente detrás de las almohadillas se interpretará y formateará como título de la sección. Incluir un espacio en blanco entre las almohadillas y el título es opcional.
- Tras el texto del título se pueden incluir también almohadillas, que serán ignoradas al procesar el documento, pero contribuyen a mejorar su legibilidad. Por ejemplo:

```
## Este es mi título ##
```

Este ejemplo es significativo de lo que significa “legibilidad” del lenguaje de marcas. A fin de cuentas un título consiste en una línea que destaca visualmente del resto del texto. Eso, en el documento ya formateado, se consigue mediante recursos tipográficos (tipo de letra, tamaño, negrita...) pero en el documento original, en texto plano, una línea que aparece entre almohadillas destaca del resto mucho más que si sólo lleva las almohadillas a su izquierda.

Es importante tener en cuenta que para que markdown interprete correctamente el carácter “#” como indicativo del inicio de una sección, es preciso que se trate de la primera línea del documento o, en caso contrario, que la línea inmediatamente anterior esté en blanco.

Para indicar los dos primeros niveles de sección, hay en markdown un método alternativo al del uso del carácter “#” consistente en subrayar el título de la sección, de manera que:

- Si el subrayado se hace con caracteres “=” el texto se interpretará como sección de nivel 1.
- Si el subrayado se hace con caracteres “-” el texto se interpretará como sección de nivel 2.

Por último, debe tenerse en cuenta que pandoc al formatear las secciones no las numera. Para cambiar este comportamiento por defecto hay que usar la opción `-N` en la línea de comandos.

### 2.1.2. Identificadores de la sección para referencias cruzadas

Una referencia cruzada implica que en una parte del texto se hace referencia a otra parte del mismo, de tal manera que al formatear el texto la referencia se ajusta. Esto se hace mediante etiquetas o identificadores únicos, es decir: una secuencia de caracteres que se asigna como nombre a una sección concreta de tal modo que ninguna otra sección tiene el mismo nombre.

Pues bien: como uno de los elementos que más se usan para referencias cruzadas son, precisamente, los títulos de las secciones, pandoc atribuye automáticamente a cada sección en el documento un identificador único que podemos usar para referencias cruzadas.

El identificador se crea a partir del título siguiendo los siguientes pasos:

- Se eliminan:
  - Todos los caracteres iniciales que no sean una letra.
  - Todos los signos de puntuación, salvo “.”, “-” y “\_”.
- Se convierte el título a minúsculas.
- Se reemplazan los espacios en blanco y saltos de línea por guiones.

Si con las partes eliminadas no quedara texto, se usa como identificador la palabra “section”

Si el nombre del identificador ya existe, se le añade un número precedido de un guión. Es decir: a la segunda aparición del mismo identificador se le añade el texto “-1”, a la tercera “-2”, etc.

Para hacer referencia a una sección usaremos entonces el siguiente formato:

`[Texto] [#identificador]`

donde Texto es el texto de la referencia que aparecerá en nuestro documento, e identificador es el identificador de la sección a que queremos referirnos.

El hecho de que los identificadores para referencias cruzadas se generen automáticamente tiene, por otra parte, un inconveniente, y es que si en algún momento dado modificamos el documento, el identificador de una concreta sección puede cambiar. Por ello cuando en un documento se hacen referencias cruzadas de este tipo conviene asegurarse, al final, de que todas las referencias sean correctas.

Este sistema de referencias cruzadas sólo funciona si la salida es en formato html, latex o context.

## 2.2. Comentarios

Un comentario es un texto que se incluye en el documento fuente, pero que luego no debe ser mostrado en el fichero de salida. Su uso es bastante útil y cuando

uno se acostumbra a ellos le cuesta usar herramientas que no los permiten (como los procesadores de texto al uso: Microsoft Word, open office, y similares).

Markdown usa, como marca de comentario, la misma que html y xml: `<!--` para iniciar el comentario y `-->` para terminarlo: todo el texto encerrado entre estas dos marcas será ignorado en la salida.

Los comentarios se pueden usar para muchas cosas:

- Para aclarar el significado de una parte del documento, o proporcionar información adicional que, en principio, sólo veremos nosotros; por ejemplo: para indicar de dónde hemos tomado ciertas ideas, o si pensamos que algo debe mejorarse ...
- Para inhabilitar temporalmente ciertas partes del documento convirtiéndolas en un comentario.
- En el caso concreto de markdown una línea no indentada que contenga exclusivamente un comentario sirve para indicar explícitamente el final de una lista (véase más adelante).

## 2.3. Formatos de carácter

Los caracteres que deben recibir un formateo especial se deben encerrar entre marcas indicativas del formato de que se trate. Los formatos de carácter que markdown reconoce son:

- **Cursiva:** El texto en cursiva debe ir encerrado entre asteriscos (\*) o signos de subrayado (\_), y tras el asterisco o signo de subrayado no puede haber un espacio en blanco.

Normalmente da absolutamente igual usar “\*” o “\_” como delimitador del formato; pero si el carácter inmediatamente anterior al delimitador inicial no es un espacio en blanco, salto de línea o marca de tabulador, hay que usar el asterisco pues “\_” sería, en este caso, interpretado literalmente.

- **Negrita:** Exactamente igual que la cursiva, pero usando como delimitadores no un asterisco o signo de subrayado sino dos asteriscos (\*\*) o dos signos de subrayado (\_\_).
- **Tachado:** El delimitador es una doble tilde (~~)
- **Subíndice:** El delimitador es una tilde simple (~)
- **Superíndice:** el delimitador es un acento circunflejo (^)

- **Verbatim:** Este formato se usa sobre todo en textos que hablan de informática para reproducir fragmentos de código o textos que hay que teclear. Normalmente se formatea usando una fuente no proporcional del tipo máquina de escribir. El delimitador para introducir este tipo de texto es el acento grave (‘) o el doble acento grave (“); este último delimitador se usa cuando el texto verbatim contiene a su vez un acento grave, para evitar que dicho acento sea interpretado como fin del texto verbatim.
- **Texto matemático:** Muchos de los lenguajes de salida manejados por pandoc, tienen algún procedimiento especial para formatear las fórmulas matemáticas. Si queremos que así se haga debemos usar como delimitador el signo del dolar (\$). En este caso los resultados dependerán mucho del formato de salida.

## 2.4. Formatos de párrafo

### 2.4.1. Párrafos normales

Un párrafo es siempre un fragmento de texto delimitado por líneas en blanco, es decir: líneas vacías o en las que sólo hay espacios en blanco o tabuladores. Es, por otra parte, indiferente, cuántas líneas en blanco hay antes o después del párrafo, siempre y cuando haya, al menos, una.

El primer párrafo puede, no obstante, empezar en la primera línea; y el último párrafo terminar en la última línea. Obsérvese que en estos dos casos en sentido estricto el párrafo no está encerrado entre líneas en blanco.

### 2.4.2. Listas

**Listas en general** Una lista es un conjunto de párrafos cada uno de los cuales constituye un elemento de la lista. El elemento de la lista tienen un identificador que puede ser un número o letra (listas numeradas) un símbolo (listas no numeradas) o un texto (listas de definiciones): el párrafo recibe una indentación especial para destacar el identificador.

Habitualmente cada elemento de la lista se incluye en un párrafo (delimitado por líneas en blanco), pero también es posible construir una lista compacta, formada por líneas y no por párrafos. Estas listas se formatean habitualmente con menor separación entre los elementos, aunque eso, claro está, depende del formato de salida.

Las listas empiezan cuando se introduce el primer elemento, y terminan cuando se introduce un párrafo que no es elemento de la lista. Puede ser un párrafo normal, un encabezamiento de sección. . .

Hay ocasiones en las que interesa indicar explícitamente el fin de una lista. Para ello no hay ninguna marca especial, pero puede usarse un truco consistente en incluir un párrafo no indentado que contenga exclusivamente un comentario.

**Clases de listas** Markdown admite las siguientes clases de listas:

**Listas desordenadas:** Para introducirlas se pueden usar los caracteres “\*”, “+”, “-”, al principio de una línea, seguidos de un espacio en blanco.

**Listas ordenadas:** Se trata de listas numeradas. Un elemento de una lista ordenada puede empezar con:

- Un número decimal, un número romano, o una letra minúscula, seguidos de un punto o un cierre de paréntesis, y un espacio en blanco.
- Una letra, mayúscula seguida de un punto y DOS espacios en blanco.
- Una letra mayúscula seguida de un cierre de paréntesis y un espacio en blanco.
- Un número decimal o romano, o una letra mayúscula o minúscula, encerrados entre paréntesis y seguidos de un espacio en blanco.
- Un carácter de almohadilla (#), seguida de un punto y un espacio en blanco. El punto y el espacio en blanco son imprescindibles para que pandoc no confunda el elemento de la lista con una sección de nivel 1.

En las listas numeradas pandoc toma en consideración la letra o número inicial de la lista; pero una vez iniciada esta, el resto de los elementos son numerados secuencial y automáticamente. Es decir: Si una lista empieza por 1, el segundo elemento necesariamente será un 2, aunque en el texto hayamos escrito otro número al principio de dicho elemento.

**Listas de definiciones:** Estas listas constan de un término seguido de su definición. Los caracteres “:” o “~” al principio de una línea, seguidos de cuatro espacios en blanco, se interpretan como inicio de una definición, por lo que se considerará automáticamente que la línea no vacía inmediatamente anterior contiene el término a definir.

En este tipo de listas hay que tener en cuenta que:

1. El término a definir no puede ocupar más de una línea
2. Si el texto de la definición ocupa más de una línea, la 2ª y posteriores líneas deben estar sangradas con, al menos, cuatro espacios en blanco.

**Listas continuadas:** El carácter especial @ al principio de una línea (encerrado o no entre paréntesis) se usa en markdown para generar un tipo de lista especial, que puede atravesar todo el documento, de tal manera que absolutamente todos los elementos de la lista sean numerados secuencialmente. Esto es útil, por ejemplo, para numerar gráficos, ejemplos, teoremas, etc.

En este tipo de listas, cualquier texto escrito inmediatamente a continuación del carácter @ será tratado como identificador del elemento de la lista, y podrá ser usado para futuras referencias cruzadas. Por ejemplo, si hemos generado una lista de ejemplos, y uno de los elementos de dicha lista empezaba con

```
(@EjemploFacil)
```

Podemos referirnos en el texto a dicho elemento en otro lugar del documento:

```
... tal y como se ilustraba en (@EjemploFacil)...
```

**Elementos de una lista que consten de más de un párrafo:** Como antes se dijo una lista termina cuando empieza un párrafo que no sea un elemento de la lista. Pero puede ocurrir que algún elemento de ésta conste de más de un párrafo. En tal caso, para indicar que dicho párrafo sigue siendo elemento de la lista, hay que indentrarlo (todo él o sólo la primera línea) con cuatro espacios en blanco o con una marca de tabulador.

OJO: Algunos editores de texto, como emacs, sustituyen automáticamente las marcas de tabulador por espacios en blanco. En este caso debemos asegurarnos de que se conserve el tabulador o de que, al menos, el número de espacios en blanco es de cuatro. Por ello yo prefiero usar directamente el espaciador, en lugar del tabulador.

**Listas anidadas:** La misma regla de los cuatro espacios (o tabulador) se usa para anidar unas listas dentro de otras, o para incluir, dentro de una lista, algún otro elemento de bloque: Los elementos anidados deben indentarse con cuatro espacios en blanco o una marca de tabulación por cada nivel de anidamiento.

El anidamiento admite varios niveles, cada uno de los cuales debe estar indentado por cuatro espacios en blanco o marca de tabulador. Así el primer nivel de anidamiento se indica con cuatro espacios, el segundo con ocho, el tercero con 12, etc.

Sobre las reglas de indentación véase el último epígrafe de este documento.

### 2.4.3. Párrafos verbatim (literales)

Un párrafo verbatim es un párrafo que normalmente se formatea con letra tipo máquina de escribir y se usa para expresar fórmulas, código informático, instrucciones literales al ordenador, y similares.

Se puede introducir mediante los siguientes procedimientos:

- Indentando todo el párrafo con un tabulador o con cuatro espacios en blanco. En este caso la indentación no se considera parte del párrafo y es suprimida al formatear.
- Encerrando el texto entre dos líneas compuestas de tres o más tildes (~) o acentos graves (`). En estos casos es indiferente la extensión de la línea inicial (siempre que sean al menos tres tildes o acentos) pero la línea de cierre no puede tener una longitud inferior a la línea inicial del bloque.

Sobre las reglas de indentación véase el último epígrafe de este documento.

### 2.4.4. Blockquotes (citas)

Se suele llamar “blockquote” o “block quotation” a un párrafo que se formatea con un sangrado especial y que se suele usar para realizar citas literales, de ahí que en español se le denomine, a veces, como “párrafo de citas”.

Para indicar un párrafo de este tipo basta con usar el carácter “>” al principio de cada una de las líneas del párrafo o, si se prefiere, al principio sólo de la primera línea del párrafo; aunque es preferible el primer procedimiento, porque insertando “>” sólo en la primera línea, se dificulta el anidar algo dentro del blockquote.

Un blockquote puede contener cualquier tipo de texto: párrafos normales, listas, o incluso otros blockquotes. Así, para empezar una lista numerada dentro de un blockquote la línea debería empezar con “> 1”, y para anidar un blockquote dentro de otro blockquote, la línea empezaría con “> >”.

## 2.5. Notas a pie de página

Markdown permite introducir notas en línea mediante la siguiente sintaxis:

```
...texto... ^[texto de la nota en línea] ...
```

Estas notas se introducen en el lugar donde debe aparecer la referencia a la nota, y ya se ocupa el sistema de numerarlas y formatearlas correctamente.

El problema que tienen estas notas en línea es que no sirven para introducir notas que tengan más de un párrafo y, además, si son muy largas, pueden hacer el texto más difícil de leer. Por ello Markdown admite un procedimiento alternativo para las notas a pie de página. En este procedimiento a cada nota se le asigna un nombre identificador, y se distingue entre la referencia a la nota propiamente dicha y el texto de la nota:

- La referencia o nombre identificador de la nota es un texto que no puede contener espacios en blanco, tabuladores ni saltos de línea. Se usa para que markdown pueda emparejar las dos partes de que consta la nota: la referencia y su texto.
- La referencia a la nota se introduce en el punto exacto del documento en el que debe aparecer el número de la nota. La referencia se hace introduciendo el nombre identificador precedido del carácter “^” y todo ello encerrado entre corchetes. Por ejemplo, si quisiéramos introducir aquí una nota llamada NotaPrueba, en nuestro documento fuente deberíamos escribir en este punto `[^NotaPrueba]`.
- El texto de la nota se puede introducir en cualquier otro punto del documento: al final, al principio, en el párrafo siguiente al de la referencia... donde más sencillo nos parezca. Para indicar que el párrafo a continuación contiene el texto de una nota, la primera línea del mismo debe empezar con `[^NombreNota]:`. Es decir: el texto de la nota llamada NotaPrueba, se escribiría en un párrafo que empezaría con: `[^NotaPrueba]:`

Si el texto de la nota consta de más de un párrafo, los párrafos a continuación del primero deben indentarse con 4 espacios en blanco o un tabulador.

## 2.6. Tablas

Markdown admite tres formas de indicar la existencia de una tabla:

### 2.6.1. Tablas simples

En las tablas simples cada fila debe ocupar una sola línea de texto, y constan de los siguientes elementos:

- La primera fila que se supone que contiene los encabezamientos de las columnas.



- Una segunda fila de control, compuesta exclusivamente de guiones que subrayarán el nombre de cada columna. Esta fila se usa para indicar que la fila anterior es el encabezado de una columna, y también la alineación de cada columna:
  - Si el subrayado coincide exactamente con el nombre de la columna esta recibirá la alineación por defecto (suele ser a la izquierda, pero puede ser justificada).
  - Si el subrayado incluye un carácter antes que el nombre de la columna, la alineación se hará a la derecha.
  - Si el subrayado incluye un carácter después que el nombre de la columna, la alineación será a la izquierda.
  - Si el subrayado incluye un carácter antes y otro después que el nombre de la columna, el texto de la columna será centrado.
- En el resto de las filas estará el contenido de la tabla. Lo que deba ir en cada columna de dicha fila deberá colocarse debajo de la fila de control, lo cual es muy difícil de conseguir si el texto no se escribe con una fuente no proporcional como, por ejemplo, courier. Las tablas escritas con fuentes proporcionales no siempre salen bien.
- La tabla debe terminar con una línea en blanco o una línea de guiones seguida de una línea en blanco
- Si la primera línea tras el fin de la tabla empieza por diciendo “Table:” el texto a continuación se interpretará como título para la tabla en cuestión.

Ejemplo de tabla simple:

Derecha	Izquierda	Centrado	Por defecto
-----	-----	-----	-----
12	33	44	55
453	225	22	4, 7

Table: Tabla de prueba en Markdown

Si las columnas no deben llevar encabezamiento, puede suprimirse esa fila, en cuyo caso la fila de control debe incluirse al principio y al final de la tabla. En este caso el alineamiento de las columnas se determinará con relación a la primera fila de la tabla.

### 2.6.2. Tablas multilínea

Las tablas multilínea van encerradas entre líneas compuestas exclusivamente de guiones y las filas de la tabla deben ir separadas por líneas en blanco. No hay que incluir línea en blanco entre el encabezado y la fila de control, ni entre ésta y la primera fila.

Hay que decir que las tablas multilínea son a veces muy imprecisas. Pandoc intenta generar una tabla en la que la anchura de las columnas sea “similar” a la anchura que éstas tenían en el documento fuente, lo cual no siempre da buenos resultados.

### 2.6.3. Tablas cuadriculadas

En la tabla cuadriculada nosotros dibujamos todos los caracteres en el documento fuente usando los siguientes ladrillos:

- Separación entre líneas: Se realiza mediante guiones salvo la separación entre la línea de encabezado y las restantes, que se hace con el carácter “=”
- Separación entre columnas dentro de una línea: En las líneas que contengan guiones o signos = se usa el carácter “+” en las que contengan texto o espacios en blanco se usa el carácter “|”.

Ejemplo de tabla cuadriculada:

+-----+-----+-----+		
Fruta	Precio	Ventajas
+=====+=====+=====+		
Naranja	3.55	Vitamina C
+-----+-----+-----+		
Coco	4.60	Extremadamente
		nutritivo
+-----+-----+-----+		

## 2.7. Otros elementos en el texto

### 2.7.1. Enlaces:

Hay varios procedimientos para crear enlaces:

- Enlaces automáticos: Basta con encerrar una url entre paréntesis agudos (< >). Por ejemplo <http://www.um.es> será convertido en un enlace a la dirección especificada.
- Enlaces en línea: Se usa el siguiente formato:

```
[Texto] (url "título")
```

donde Texto es el texto del enlace que se mostrará en el documento, la url es la dirección a la que apunta el enlace, y el texto entrecomillado —que es opcional— representa un título para el enlace. El texto del enlace puede incluir marcas de formato, pero el del título no.

- Enlaces con nombre: En el caso de enlaces que usamos con mucha frecuencia podemos definirlos asignándoles un nombre de tal modo que para usar el enlace baste con escribir dicho nombre. Para asignar un nombre a un enlace se usa el formato siguiente:

```
[Nombre]: url "título"
```

donde Nombre es el nombre con el que será identificado el enlace, url es la url a que apunta el enlace, la cual puede, opcionalmente, ir encerrada entre paréntesis agudos, y el título del enlace, que es opcional, puede ir también en la línea siguiente.

El uso de estos enlaces con nombre es similar al de las notas a pie de página: El nombre del enlace se puede asignar en cualquier parte del documento que no sea el interior de otro elemento. Para crear un enlace en cualquier punto del documento usaremos el formato

```
[Texto] [Nombre]
```

donde Texto es el texto que aparecerá en el documento, y Nombre es el nombre asignado al enlace. Si el texto que queremos que aparezca coincide con el nombre del enlace, podemos suprimir la segunda parte, o incluir corchetes vacíos.

### 2.7.2. Imágenes:

Un enlace no automático precedido inmediatamente por el carácter “!” es tratado como referencia a una imagen. La url se interpreta entonces como dirección del fichero donde se contiene la imagen.

Si un párrafo contiene exclusivamente un enlace a una imagen, se añadirá un título a la imagen, usando para ello el texto del enlace.

### 2.7.3. Bloques de título:

Es corriente en la documentación científica y técnica que las tres primeras líneas de un documento contengan, respectivamente, el título, el nombre del autor y la fecha de composición. A esta información se la denomina “bloque de título”. Markdown identifica un bloque de título si las primeras líneas del documento empiezan con el carácter “%”. En tal caso:

- La primera línea se asignará al título.
- La segunda línea se asignará al autor.
- La tercera línea se asignará a la fecha.

Por ejemplo: las tres primeras líneas de este documento, son:

```
% Sinopsis del formato markdown para pandoc
% Joaquín Ataz López
% Febrero de 2012
```

Los tres elementos informativos que componen el bloque de título son secuenciales, es decir: puede aparecer sólo el primero, o sólo el primero y el segundo, pero no puede aparecer el segundo si no hay primero, y no puede aparecer el tercero si no hay primero y segundo. Por lo tanto si queremos incluir exclusivamente información sobre la fecha (tercer elemento) deberemos incluir antes dos líneas vacías que empiecen por el carácter “%” para que pandoc identifique correctamente dónde está la fecha.

Esto es así porque la información del bloque de título, además de ser formateada y aparecer en el documento final, dependiendo del formato de salida, formará parte de la llamada “metainformación del documento” y en este sentido es importante que no se identifique como título lo que es el nombre de un autor. En HTML, por ejemplo, la etiqueta `<title>` en el encabezado del documento, se usa también como título de la ventana, y como información dada al navegador para el historial de navegación.

Tanto el título como los autores pueden incluir más de una línea, en cuyo caso el carácter “%” sólo aparecerá en la primera de las líneas, pero la segunda y sucesivas líneas deben empezar por uno o más espacios en blanco. En el caso de los autores este procedimiento puede usarse para indicar múltiples autores, dedicando una línea a cada uno de ellos; aunque también pueden indicarse todos los autores en la misma línea, separándolos por un punto y coma.

NOTA: El bloque de título sólo es procesado si pandoc es llamado con la opción “-s” o con la opción “--template” que presupone el uso de la anterior.

#### 2.7.4. Caracteres literales:

En ocasiones podemos desear introducir en el texto un carácter que tiene un significado especial para markdown, pero no queremos que markdown lo interprete del modo habitual, sino literalmente. Por ejemplo: podemos desear encerrar una palabra entre asteriscos, sin que ésta vaya a formatearse en cursiva. Para conseguir esta interpretación literal de los caracteres, hay dos vías:

- La más simple: preceder el carácter en cuestión con una barra invertida (\). Este carácter indica a markdown que el próximo carácter debe ser leído literalmente.
- Otro procedimiento es usar texto verbatim: Ya hemos visto que en markdown hay párrafos verbatim (indicados por la indentación o por estar encerrados entre líneas de tildes o acentos graves) y caracteres verbatim (encerrados entre acentos graves). Pues bien: una característica adicional del formato verbatim es que todos los caracteres que vayan en dicho formato se interpretan literalmente.

#### 2.7.5. Líneas horizontales:

Una línea de tres o más asteriscos, guiones o signos de subrayado (contiguos o separados por espacios en blanco), se convertirá en una línea horizontal que atraviese todo el ancho del párrafo.

### 2.8. Resumen de las reglas de indentación

Uno de los aspectos más confusos de markdown cuando no se tiene costumbre es el de las reglas de indentación. En particular la indentación de los cuatro espacios en blanco. Esta se usa para distintas cosas:

- (a) En las listas ordenadas o desordenadas, para indicar el segundo o ulterior párrafo de un concreto elemento de la lista.
- (b) En las listas de definiciones, para indicar que una línea forma parte de la misma definición que la línea anterior.
- (c) En las notas a pie de página, para indicar el segundo o ulterior párrafo de una nota con nombre.
- (d) En las listas anidadas para indicar niveles de indentación.
- (e) Para introducir un párrafo verbatim.

Los tres primeros usos se pueden simplificar en una sola regla: un párrafo indentado con cuatro espacios en blanco pertenece al mismo bloque y debe tener el mismo formato que el párrafo anterior.

Estos distintos usos de una misma regla pueden, por otra parte, causar a veces problemas de interpretación. Por ejemplo: si tras una lista, o nota a pie de página, queremos introducir un párrafo verbatim, los cuatro espacios en blanco que indican la presencia de dicho tipo de párrafo, serían interpretados incorrectamente. Para evitar que esto ocurra, hay dos vías:

1. Podemos introducir el párrafo verbatim mediante alguno de los procedimientos alternativos como, por ejemplo, encerrándolo entre dos líneas de tildes o de acentos graves.
2. También podemos incluir un párrafo vacío que explícitamente indique que la lista o la nota terminan ahí. Un párrafo vacío es, a estos efectos, un párrafo que, sin ninguna indentación, contiene exclusivamente un comentario que, por lo tanto, no se procesará. Es decir: un párrafo que contenga exclusivamente

<!-- -->

Si, por otra parte, queremos introducir un párrafo verbatim dentro de una lista o de una nota a pie de página, podemos:

1. Indentarlo en cuatro espacios (para que se interprete correctamente que dicho párrafo pertenece a la lista o nota) y encerrarlo entre líneas de tildes o acentos graves.
2. Indentarlo en otro espacios. Los cuatro primeros espacios se interpretarán como “pertenencia a la lista” y los cuatro siguientes como “inicio de párrafo verbatim”.

---

Aparte de la regla de los cuatro espacios, en general markdown ignora cualquier otra indentación, salvo en el caso de los bloques de título donde, como antes se dijo, si el título o el autor deben constar de más de una línea, la segunda y sucesivas líneas deben estar indentadas por, al menos, un espacio en blanco.