Usando Pandoc

Sergio Alvariño [salvari@gmail.com](mailto:salvari@gmail.com)

abril-2016

Una chuleta para usar Pandoc, copiado descaradamente de varios sitios de internet.

Solo para referencia rápida y personal.

# ¿Cómo usar esto?

## Muy rápido

Clona el repo en un directorio :

git clone https://bitbucket.org/salvari/pandoc\_basico

Renombra el directorio:

mv pandoc\_basico miProyecto

Elimina la info de git

rm -rf miProyecto/.git

Edita el fichero miProyecto/src/documento.md con tu editor de texto favorito.

Ejecuta:

make

Para generar todos los ficheros de salida y el fichero README.md (equivale a *make all*)

make clean

Para borrar todos los ficheros de salida

make reset

Equivale a *make clean all*

## Más detalles

El makefile está preparado para procesar **todos** los ficheros con extensión *.md* que haya en el directorio *src*. Esto permite escribir documentos largos y dividirlos en secciones, por ejemplo podríamos tener los siguientes documentos en el directorio *src*

00\_Comienzo.md  
10\_Capitulo\_01.md  
20\_Capitulo\_02.md  
30\_Conclusion.md  
40\_apendices.md

Al ejecutar make nos crearía **un solo documento de salida** concatenando todos los ficheros. El orden en que los concatena es el orden en el que aparecen al hacer un *ls* por eso se nombran con una numeración al principio que permita ordenarlos a gusto del autor.

Si quieres cambiar el nombre del fichero de salida (*documento*) tendrás que editar el makefile y cambiar la línea:

target := documento

Otras líneas que puedes tocar en el makefile son las que especifican el idioma y los tipos de letra usados.

# ¿Qué es Pandoc?

Como explican en http://pandoc.org, Pandoc es una librería en Haskell para hacer conversión de documentos de un formato markup a otro. Y también es una herramienta de terminal de comandos que usa esa librería.

Lo que nos permite Pandoc a la hora de documentar un proyecto es mantener la documentación en un formato abierto y sencillo (markdown) y generar salidas en distintos formatos (pdf, mediawiki, epub, html, etc) con un simple comando.

# ¿Qué necesitas tener instalado?

* Pandoc
* make
* git (no es imprescindible pero muy recomendable)
* Las plantillas de Pandoc (o *templates*)
* Un buen editor de texto

# Chuletario de Pandoc

## Backslash Escapes

Salvo que estemos dentro de un bloque de código o de "código en linea", **cualquier carácter de puntuación o espacio** precedido de contrabarra se tratará de forma literal, incluso si ese carácter normalmente indique algún formato.

## Bloque de título

Es una forma rápida de indicar el título el autor o autores y la fecha. Tiene que ir al principio del documento

% título  
% autor(es) (separados por :)  
% fecha

Alternativamente se puede usar otro estilo para el bloque de título, mucho más completo, en formato [YAML](https://en.wikipedia.org/wiki/YAML), especificando variables. No puede usarse simultáneamente con el anterior, hay que escoger entre los dos estilos.

Se pueden especificar todo tipo de variables [[1]](#footnote-31).

---  
title: Título  
author:  
- Autor Uno <autor.uno@correo.com>  
- Otro autor <otroautor@correo.com>  
tags: [nothing, nothingness]  
date: enero-2016  
lang: es-ES  
abstract: |  
 Este es el resumen.  
  
 Con dos párrafos.  
...  
---

## Incrustar TeX y HTML

* Los comandos TeX se pasan de forma transparente al Markdown, y afectan solo a la salida de LaTeX y ConTeXt; en el resto de casos se borran
* El código HTML pasará a la salida sin cambios, pero el Markdown dentro de los bloques HTML se procesa como Markdown

## Párrafos y retornos de línea

* Un párrafo es una o más líneas de texto separadas por una linea en blanco del resto
* Una línea que termina con dos espacios, o una línea que termina con un fin de linea escapado (contrabarra seguida de retorno de linea) indica un cambio de linea manual

## Itálica, negrita, superescrito, subesctrito, tachado

\*Itálica\* and \*\*negrita\*\* se indican con asteriscos.  
  
Para ~~tachar~~ texto usa tildes dobles.  
  
Superscrito se indica así: 2^ndo^.  
  
Subescrito con tildes simples, así: H~2~O.  
  
Los espacios en el superescrito y el subescrito tienen que ir escapados,  
p.ej., H~esto\ es \ un\ subescrito~.

## TeX matématico o código incrustado en linea

El TeX matemático va entre signos$: $2 + 2$.  
  
El código en linea va entre comillas invertidas: `echo 'hello'`

## Enlaces e imágenes

<http://example.com>  
<foo@bar.com>  
[inline link](http://example.com "Title")  
![inline image](/path/to/image, "alt text")  
  
[reference link][id]  
[implicit reference link][]  
![reference image][id2]  
  
[id]: http://example.com "Title"  
[implicit reference link]: http://example.com  
[id2]: /path/to/image "alt text"

## Notas al pie de página

Las notas en linea son como  
esta.^[Nótese que las notas en linea no pueden tener más de un párrafo.]  
Las notas de referencia son como esta.[^id]  
  
[^id]: Las notas de referencia pueden contener varios párrafos.  
  
 Los parámetros a continuación deben estar identados.

## Citas

Blah blah [see @doe99, pp. 33-35; also @smith04, ch. 1].  
  
Blah blah [@doe99, pp. 33-35, 38-39 and \*passim\*].  
  
Blah blah [@smith04; @doe99].  
  
Smith says blah [-@smith04].  
  
@smith04 says blah.  
  
@smith04 [p. 33] says blah.

## Encabezados

Encabezado 1  
========  
  
Encabezado 2  
--------  
  
# Encabezado 1 #  
  
## Encabezado 2 ##

Las almohadillas de cierre # son opcionales. Es necesario añadir una línea en blanco antes y después de cada cabecera.

## Listas

#### Listas Ordenadas

1. example  
2. example  
  
A) example  
B) example

#### Listas desordenadas

Los items de la lista deben ir marcados con '\*', '+', or '-'.

+ example  
- example  
\* example

Las listas se pueden anidar de la forma usual:

+ example  
 + example  
+ example

#### Listas de definición

Term 1  
  
: Definition 1  
  
Term 2  
  
: Definition 2  
 Second paragraph of definition 2.

## Blockquotes

> blockquote  
>> nested blockquote

Es necesario añadir lineas en blanco antes y después de los bloques-cita.

## Tablas

Right Left Center Default  
------- ------ ---------- -------  
 12 12 12 12  
 123 123 123 123  
 1 1 1 1  
  
Table: Demonstration of simple table syntax.

(Para tablas más complejas consulta la [documentación de Pandoc](http://pandoc.org/README.html#tables).)

## Bloques de código

Los bloques de código empiezan con tres o más tildes; y acaban por lo menos con el mismo número de tildes:

~~~~~~~  
{code here}  
~~~~~~~

Opcionalmente, se puede especificar el lenguaje que corresponde al bloque de código:

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ {.haskell .numberLines}  
qsort [] = []  
qsort (x:xs) = qsort (filter (< x) xs) ++ [x] ++  
 qsort (filter (>= x) xs)  
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

## Lineas horizontales

3 o mas guiones o asteriscos en una linea (se permiten espacios intercalados)

---  
\* \* \*  
- - - -

## Bloques verbatim

Todo el texto identado cuatro espacios

Ejemplo Esto es un bloque verbatim y por ejemplo \*esto\* aparece  
tal cual y no en itálica.

# En que me he basado (o copiado si lo prefieres)

* En la [guia de usuario de Pandoc](http://pandoc.org/README.html) Importante leersela para sacarle todo el jugo a esta herramienta
* En la [chuleta de Pandoc](https://github.com/dsanson/Pandoc.tmbundle/blob/master/Support/doc/cheatsheet.markdown) de [David Sanson](https://github.com/dsanson), perfecta para referencia rápida
* Para hacer el makefile me he leido varios tutoriales y copiado descaradamente de varios sitios que olvidé apuntar (lo siento)

1. Ojo por que en el makefile propuesto se especifica el lenguaje, asi que la variable del bloque de título no va a tener efecto en este caso. [↑](#footnote-ref-31)