

Title: markdown to pdf Date: 2017-1-27 12:12 Modified: 2017-1-27 12:12 Category: Software Tags: markdown, pdf, pandoc, gimli, linux, atom, markdownpad, cloudconvert, markdowntopdf Authors: procamora Slug: markdown-to-pdf Summary: A la hora de crear un fichero pdf a través de un fichero markdown tenemos múltiples opciones, las mas personalizada es *pandoc* aunque otros conversores también son muy personalizables. Status: published

Introducción:

A la hora de crear un fichero pdf a través de un fichero markdown tenemos múltiples opciones, las mas personalizada es *pandoc* aunque otros conversores también son muy personalizables.

Tenemos distintos tipos de conversores de markdown a pdf:

- Conversores por linea de comandos:
 - pandoc
 - gimli
- Conversores con entorno gráfico:
 - Atom
 - Markdown Pad
- Conversores Online:
 - Cloudconvert
 - Markdowntopdf
- Resumen general
- Conclusion final

Puede asegurarse de que ciertas cosas, como las comillas y los guiones en em, se leen y formateen correctamente especificando el conmutador “Smart” (un mayúscula -S o -smart):

Conversores por linea de comandos

- pandoc
- gimli

Pandoc

Pandoc es una herramienta que te permite a través de un fichero markdown generar otro con múltiples formatos, como por ejemplo PDF, HTML o DOCX. Los pasos a realizar son los siguientes:

Mejora a conseguir: que cada inicio de apartado empiece en una nueva pagina

Paso 1: Instalación.

La instalación en Fedora es muy simple:

```
dnf install pandoc pandoc-citeproc texlive texlive-collection-langspanish.noarch
```

Paso 2: Añadir cabecera yaml al fichero markdown.

Una vez instalado pandoc mas todas las librerías necesarias de texlive con latex tendremos que añadir una cabecera yaml para indicarle a pandoc una serie de directrices a la hora de generar el pdf.

Los parámetros a tener en cuenta son: - geometry: modificamos la alineación aumentándola - toc: Indicamos con true que queremos una tabla de contenidos - documentclass: pone en negrita el texto del los metadatos, visualmente queda mejor

MODIFICAR

```
---
title: Nombre de la practica
subtitle: Asignatura
author:
- nombre 1
- nombre 2
date: September 6, 2016
```

```

header: dsad
footer: So is this
geometry: margin=1in
toc: true
documentclass:
- scrartcl
---
```

Paso 3: Creación del fichero plantilla de latex para poner un pagina en blanco antes de la tabla de contenidos y después.

1. Generar la plantilla por defecto de latex, es recomendable generarla para que tenga la versión mas reciente usada por pandoc, la que yo uso es esta.

```
pandoc -D latex > plantilla.tex
```

2. Modificar la plantilla, buscando donde se crea la tabla de contenido para añadir las paginas en blanco. Fuente.

Buscar este texto

```

$if(toc)$
{
$if(colorlinks)$
\hypersetup{linkcolor=$if(toccolor)$$toccolor$$else$black$endif$}
$endif$
\setcounter{tocdepth}{$toc-depth$}
\tableofcontents
}
```

Y le añadimos los dos **

** para dejar la tabla de paginas en una pagina sola.

```
$if(toc)$
{
  \newpage
  $if(colorlinks)$
  \hypersetup{linkcolor=$if(toccolor)$$toccolor$$else$black$endif$}
  $endif$
  \setcounter{tocdepth}{$toc-depth$}
  \tableofcontents
  \newpage
}
```

3. Latex dispone de 2 tags interesantes `\newpage` y `\pagebreak` que nos servirán para terminar de maquetar los texto como nosotros queramos, poniendo cualquiera de los 2 tags en el fichero markdown podremos introducir un final de pagina, por lo que si los ponemos antes de cada nuevo apartado, tendremos cada apartado en una pagina diferente. Fuente

4. Añadir bibliografía: hay que crear la bibliografía que tendra este formato:

Fichero bibliografia.bib:

```
@article{fenner2012a,
  title = {One-click science marketing},
  volume = {11},
  url = {http://dx.doi.org/10.1038/nmat3283},
  doi = {10.1038/nmat3283},
  number = {4},
  journal = {Nature Materials},
  publisher = {Nature Publishing Group},
  author = {Fenner, Martin},
  year = {2012},
  month = {mar},
  pages = {261-263}
}
```

Codigo del .md

This is a test2 [@fenner2012a].

Despues se compilara con: `pandoc --bibliography=bibliografia.bib`

Paso 4: Generación del pdf.

Teniendo esa cabecera yaml en el fichero y generada la plantilla modificada para general el pdf con nuestro formato ya solo falta llamar al comando `pandoc`.

Los parámetros a tener en cuenta son: `- --template:`

Otros parámetros interesantes: `- --number-sections:`

El comando final resultante seria:

```
pandoc --template=plantilla.tex memoria.md -o output.pdf
```

Opción 2: En ocasiones el código a mostrar tiene líneas muy largas y `pandoc` no las parte adecuadamente, hay una opción para hacer que markdown corte las líneas de código pero tiene ciertas desventajas como son: no resalta el código y necesita el motor de `xelatex`

Para usar esta opción solo habría que añadir un nuevo parámetro a la cabecera, que sera `header-includes` quedando así la cabecera yaml:

```
---
title: Recorrido de un paquete
subtitle: Redes de Comunicaciones
author:
```

```

- Pablo José Rocamora Zamora
;date: Enero 30, 2017
header: dsad
footer: So is this
geometry: margin=1in
toc: true
documentclass:
- scrartcl
header-includes:
- \usepackage{xcolor}
- \lstset{breaklines=true}
- \lstset{basicstyle=\small\ttfamily}
- \lstset{extendedchars=true}
- \lstset{tabsize=2}
- \lstset{columns=fixed}
- \lstset{showstringspaces=false}
- \lstset{frame=trbl}
- \lstset{frameround=tttt}
- \lstset{framesep=4pt}
- \lstset{numbers=left}
- \lstset{numberstyle=\tiny\ttfamily}
- \lstset{postbreak=\raisebox{0ex}[0ex][0ex]{\ensuremath{\color{red}\hookrightarrow\space}}}}
- \lstset{keywordstyle=\color{rgb}{0.13,0.29,0.53}\bfseries}
- \lstset{stringstyle=\color{rgb}{0.31,0.60,0.02}}
- \lstset{commentstyle=\color{rgb}{0.56,0.35,0.01}\itshape}
- \lstset{stepnumber=1}
- \lstset{numbersep=5pt}
- \lstset{backgroundcolor=\color{RGB}{248,248,248}}
- \lstset{showspaces=false}
- \lstset{showtabs=false}
- \lstset{captionpos=b}
- \lstset{breakatwhitespace=false} # esto hace que sea una linea son contar
- \lstset{breakautoindent=true}
- \lstset{escapeinside={\%*}{*}}
- \lstset{linewidth=\textwidth}
- \lstset{basewidth=0.5em}

```

Hay muchos mas parametros que podemos poner con lstset, aqui podemos ver el **listado con listings** muchos:

Si no tenemos el motor de **xelatex** instalado tendremos que instalarlo antes de poder usar el comando: Fuente

```
[sudo] dnf install texlive texlive-latex texlive-xetex texlive-collection-latex texlive-collection-latexrecom
```

Instalar solo si es necesario

```
[sudo] dnf install texlive-collection-latexextra
```

El comando final resultante seria:

```
pandoc --template=plantilla.tex --listings --latex-engine=xelatex memoria.md -o memoria.pdf
```

			resaltado		Separar		
			de	tablas	por pagina		
Portada	Indice	tabla de	sintaxis	markdown	distintos	Optimización	Corta lineas de
		contenidos			apartados	del espacio	cdigo largas

Gimli

Gimli es una herramienta para convertir ficheros markdown en pdf, esta escrita en ruby. Su instalación se hace a través del instalador de paquetes de ruby **gem**. Trabaja convirtiendo el markdown a html a través de `github/markup` y el html a pdf a través de `wkhtmltopdf`, por lo que gracias a `wkhtmltopdf` podremos hacer pdf con bastantes mejoras visuales.

Paso 1: Instalación.

En Fedora 24 viene instalado por defecto **gem**, así que la instalación es bastante simple:

```
$ [sudo] gem install gimli
```

Paso 2: Generación del pdf.

Para generar un pdf con gimli, simplemente habría que usar el comando `gimli -f memoria.md` y generaría un fichero pdf con el mismo nombre aunque con extensión pdf, por defecto el fichero que genera es bastante feo estéticamente, pero le podemos añadir varias mejoras visuales a través de parámetros que enviamos a `wkhtmltopdf` directamente gracias al parámetro `-w`:

Gimli tiene un parámetro de configuración que es `-cover portada.md` que es para ponerle una portada al pdf resultante.

Parametros de wkhtmltopdf: Todos estos parámetros tienen que pasarse a través del parámetro `-w` de gimli para que lleguen a `wkhtmltopdf`, por lo que el comando seria `gimli -f test.md -w 'PARÁMETROS'`

- Tabla de contenidos: tendremos que pasar el parámetro `--toc` para generar la tabla y con `--toc-header-text "Tabla de contenidos"` indicamos el nombre que tendra, quedando así el comando resultante:

```
gimli -f test.md -w '--toc --toc-header-text "Tabla de contenidos"'
```

- Pie de pagina con numero: tendremos que pasar el parámetro `--footer-right "[page]/[toPage]"`, aunque también podemos cambiar la posición del texto con: `--footer-left "[page]/[toPage]"` o `--footer-center "[page]/[toPage]"` quedando así el comando resultante:

```
gimli -f test.md -w '--footer-right "[page]/[toPage]"'
```

Como resultado final nos quedaría este comando para generar un pdf con tabla de contenidos mas pie de pagina con el numero

```
gimli -f test.md -cover portada.md -w '--toc --toc-header-text "Tabla de contenidos" --footer-right "[page]/[toPage]"'
```

		resaltado		Separar por pagina			
Portada	Indice	tabla de contenidos	de sintaxis	tablas markdown	distintos apartados	Optimización del espacio	Corta lineas de código largas
*							

* Puede generar una portada pero tiene que estar hecha en markdown y no suelen quedar muy bien visualmente.

Conversores con entorno gráfico

- Atom
- Markdown Pad

Atom

Atom por si mismo no tiene ningún conversor de markdown a pdf, pero tiene varios paquetes que desempeñan esa tarea con bastante calidad. El mas destacado es `markdown-themeable-pdf` que crea unos pdf con bastantes características interesantes.

Una vez instalado con pulsar `ctrl-shift-E` o botón derecho > **Markdown to PDF** generaremos el pdf con el mismo nombre que el fichero.

Se puede modificar el css, el pie de pagina y la cabecera resultante, los ficheros a modificar serian:

- ~/atom/markdown-themeable-pdf/styles.css
- ~/atom/markdown-themeable-pdf/header.js
- ~/atom/markdown-themeable-pdf/footer.js

ACORDARME DE SUBIR MIS FICHEROS MODIFICADOS A CODE

			resaltado		Separar		
		tabla de	de	tablas	por pagina	Optimización	Corta lineas de
Portada	Indice	contenidos	sintaxis	markdown	distintos apartados	del espacio	codigo largas

Markdown Pad

Markdown Pad es el único de todos los conversores que salen aquí que esta unicamente para Windows y que ademas es de pago, teniendo una versión gratuita bastante limitada.

PONERLO CUANDO ESTE EN WINDOWS YA QUE NO SE PUEDE USAR EN LINUX

			resaltado		Separar		
		tabla de	de	tablas	por pagina	Optimizacion	Corta lineas de
Portada	Indice	contenidos	sintaxis	markdown	distintos apartados	del espacio	codigo largas

Conversores Online

Los conversores online están muy bien a la hora de generar un documento rápidamente, pero incorporan muchas menos características que el resto de conversores.

Cloudconvert

url

			resaltado		Separar		
		tabla de	de	tablas	por pagina	Optimización	Corta lineas de
Portada	Indice	contenidos	sintaxis	markdown	distintos apartados	del espacio	código largas

Markdowntopdf

			resaltado		Separar		
		tabla de	de	tablas	por pagina	Optimización	Corta lineas de
Portada	Indice	contenidos	sintaxis	markdown	distintos apartados	del espacio	código largas

<http://www.markdowntopdf.com/>

Resumen general

Programa	Portada	Indice	tabla de contenidos	resaltado de sintaxis	tablas markdown	Separar por pagina distintos apartados	Optimización del espacio	Corta líneas de código largas
Pandoc								
Gimli	*							
Atom								
Markdown Pad								
Cloudconvert								
Markdown to pdf								

* Puede generar una portada pero tiene que estar hecha en markdown y no suelen quedar muy bien visualmente.

Conclusión final

Creo que en este caso la mejor opción es pandoc ya que es la único que puede hacer todo con un mínimo esfuerzo, como por ejemplo la portada y la toc.