

# Breve tutorial para redactar artículos científicos

---

## Objetivos

En este tutorial aprenderás lo básico de Markdown -una sintaxis de marcado para texto plano que es fácil de leer y de escribir-, así como [Pandoc](#), una herramienta de línea de comandos que convierte el texto plano en varios tipos de archivos bellamente formateados: PDF, .docx, HTML, LaTeX, presentaciones de diapositivas y más.<sup>1</sup> Con Pandoc como tu herramienta digital de composición tipográfica, puedes usar la sintaxis de Markdown para añadir figuras, una bibliografía, formato, y cambiar fácilmente estilos de citación de Chicago a MLA (por ejemplo), todo ello utilizando texto plano.

El tutorial asume que no tienes conocimientos técnicos previos, pero escala con la experiencia ya que a menudo sugerimos técnicas más avanzadas hacia el final de cada sección. Éstas están claramente marcadas y pueden ser revisitadas después de alguna práctica y experimentación.

En vez de seguir este tutorial de una manera mecánica, te recomendamos esforzarte por entender las soluciones ofrecidas aquí como una *metodología* que necesitarías adaptar posteriormente para ajustarla a tu entorno y flujo de trabajo. La instalación de las herramientas necesarias presenta tal vez el mayor obstáculo para la participación. Destina suficiente tiempo y paciencia para instalar todo correctamente o házlo con un colega que tenga una configuración similar para ayudarse mutuamente.

## Filosofía

Escribir, almacenar y recuperar documentos son actividades centrales en el flujo de trabajo de la investigación en humanidades. Sin embargo, muchos autores basan su práctica en herramientas y formatos propietarios que a veces no cubren ni siquiera los requerimientos básicos de la escritura académica. Habrás experimentado cierta frustración por la fragilidad de las notas a pie de página, las bibliografías, figuras y borradores de libros escritos en Microsoft Word o Google Docs. Sin embargo, la mayoría de las revistas aún insisten en recibir textos en formato .docx.

Pero más que causar una frustración personal, esta dependencia a las herramientas y formatos propietarios tiene implicaciones negativas a largo plazo para la comunidad académica. En este entorno, las revistas deben subcontratar la composición tipográfica, alienan a los autores de los contextos materiales de la publicación añadiendo otros obstáculos innecesarios a la libre circulación del conocimiento.<sup>3</sup>

Cuando utilizas MS Word, Google Docs u Open Office para escribir documentos, lo que ves no es lo que obtienes. Debajo de la capa visible de palabras, oraciones y párrafos se encuentra una complicada capa de código comprensible solamente para las máquinas. Debido a esta capa oculta, tus archivos `.docx` y `.pdf` dependen de herramientas propietarias para mostrarse correctamente. Dichos documentos son difíciles de buscar, de imprimir y de convertir a archivos con otros formatos.

Más aún, el tiempo utilizado en formar el documento en MS Word u Open Office se desperdicia, porque el editor de la revista donde lo envías retira todo ese formato. Tanto los autores como los editores se beneficiarían si intercambiaban archivos con un formato mínimo, dejando la composición tipográfica a la etapa de composición final del proceso de publicación.

Aquí es donde brilla Markdown. Markdown es una sitaxis para el marcado semántico de elementos dentro de un documento de forma explícita, no en alguna capa oculta. La idea es identificar las unidades que tienen significado para los seres humanos como títulos, secciones, subsecciones, notas a pie de página y las ilustraciones. Por lo menos, los archivos seguirán siendo comprensibles para ti, incluso si el editor de textos que estás utilizando deja de funcionar o si queda fuera del mercado.

Escribir en esta forma libera al autor de la herramienta. Markdown se puede escribir en cualquier editor de texto y ofrece un rico ecosistema de *software* que puede representar ese texto en documentos con aspecto atractivo. Por esta razón, Markdown está experimentando un periodo de crecimiento, no solamente como un medio para la escritura de documentos académicos sino como una convención para la edición en línea en general.

Los editores de texto para todo propósito más populares incluyen [Atom](#) (para todas las plataformas) y [Notepad++](#) (para Windows).

Es importante entender que Markdown no es más que una convención. Los archivos Markdown se almacenan como texto plano, además de añadir la flexibilidad del formato. Los archivos de texto plano han existido desde los tiempos de las máquinas de escribir electrónicas. La longevidad de este estándar hace, de manera inherente, que sean más sostenibles y más estables que los formatos propietarios. Mientras que los archivos producidos hace diez años en Microsoft Word o en Pages de Apple pueden causar serios problemas cuando se abren con la última versión del programa, aún es posible abrir un

archivo de texto plano escrito en alguno de los editores de texto “muertos”, del pasado, muchas décadas después: AlphaPlus, Perfect Writer, Text Wizard, Spellbinder, WordStar o SCRIPSIT2.0, el favorito de Isaac Asimov producido por Radio Shack. Escribir en texto plano te garantiza que tus archivos permanecerán legibles diez, quince o veinte años a partir de ahora. En esta lección se describe un flujo de trabajo que libera al investigador de programas de procesamiento de texto propietarios y archivos de formatos frágiles.

Ahora es posible escribir una amplia gama de documentos en un formato -artículos, entradas de blogs, wikis, programas de estudio y cartas de recomendación-, utilizando el mismo conjunto de herramientas y técnicas para buscar, descubrir, hacer copias de seguridad y distribuir nuestros materiales. Tus notas, entradas de blog, documentación de código y wikis pueden ser creados en Markdown. Cada vez más, muchas plataformas como WordPress, Reddit y GitHub soportan nativamente la escritura en Markdown. A largo plazo, tu investigación se beneficiará de este tipo de flujos de trabajo unificados, lo que hace que sea más fácil guardar, buscar, compartir y organizar tus materiales.

## Principios

Inspirados en las buenas prácticas de una variedad de disciplinas nos hemos guiado por los siguientes principios:

1. *Sostenibilidad.* El texto plano a la vez garantiza transparencia y responde a las normas de conservación a largo plazo. MS Word puede seguir el camino de Word Perfect en el futuro, pero el texto plano seguirá siendo siempre fácil de leer, catalogar, minar y transformar. Por otra parte, el control de versiones de texto plano permite efectuar cambios de manera fácil y potente, lo cual es muy útil en la creación colaborativa y organización de borradores. Tus archivos de texto plano serán accesibles en teléfonos celulares, tabletas, o tal vez en una terminal de baja potencia de alguna biblioteca lejana. El texto plano es compatible con versiones anteriores y tiene garantía de futuro. Cualquier *software* o *hardware* que se presente más adelante será capaz de entender tus archivos de texto plano.
2. *Preferencia por formatos legibles por humanos.* Al escribir en Word o en Google Docs, lo que se ve no es lo que se obtiene. El archivo .docx tiene caracteres escondidos, generados automáticamente, que crean una capa de composición tipográfica eclipsada que hace que al usuario le sea difícil solucionar problemas. Algo tan sencillo como pegar una imagen o un texto desde el navegador puede tener efectos impredecibles en el formato del documento.
3. *Separación de forma y contenido.* Escribir y formar al mismo tiempo distrae. La idea es escribir primero y dar formato más tarde, lo más cerca posible al momento de la publicación. Una tarea como cambiar el estilo de citación de Chicago a MLA debe ser posible sin esfuerzo. Los editores de revistas que quieran ahorrar tiempo ante formatos innecesarios y corrección de textos, deben ser

capaces de proporcionar a sus autores una plantilla de formato que se encargue de las minucias de la composición tipográfica.

4. *Soporte del aparato crítico*. El flujo de trabajo tiene que manejar con gracia notas a pie de página, cifras, caracteres internacionales y bibliografía.
5. *Independencia de plataforma*. Como las plataformas de publicación se multiplican, tenemos que ser capaces de generar una multiplicidad de formatos, incluyendo presentaciones de diapositivas, impresión, web y dispositivos móviles. Idealmente, nos gustaría ser capaces de generar los formatos más comunes sin romper las referencias bibliográficas. Nuestro flujo de trabajo debe ser portátil al grado que sería bueno poder copiar una carpeta a un *pendrive* y saber que contiene todo lo necesario para su publicación. Escribir en texto plano significa que puedes compartir fácilmente, editar y archivar tus documentos en prácticamente cualquier entorno. Por ejemplo, un temario de clase escrito en Markdown puede ser guardado como PDF, impreso como hoja de mano, convertido a HTML para la Web, todo desde el mismo archivo. Los archivos impresos y subidos a la web pueden ser publicados de la misma fuente y tener un aspecto similar, preservando la distribución lógica del material.

Markdown y LaTeX responden a todas estas exigencias. Elegimos Markdown (y no LaTeX) porque ofrece la sintaxis más ligera y libre de desorden (de ahí “*mark down*”), y porque cuando se combina con Pandoc permite una mayor flexibilidad de salidas (incluyendo archivos `.docx` y `.tex`).<sup>4</sup>

## Requisitos de software

Expresamente omitiremos algunos detalles menudos relacionados con la instalación del *software* listado abajo para cada plataforma o sistema. Por ejemplo, no tiene sentido proporcionar las instrucciones de instalación para LaTeX cuando las instrucciones en línea para tu sistema operativo siempre serán más completas y actualizadas. De la misma manera la mecánica de la instalación de Pandoc se obtiene de manera más completa si buscas en Google “installing Pandoc”, con lo que probablemente el primer resultado sea la página principal de Pandoc.

- **Editor de Texto Plano.** Entrar al mundo de la edición en texto plano amplía tu capacidad de seleccionar herramientas innovadoras de manera espectacular. Busca en línea “markdown text editor” y experimenta con las opciones. No importa lo que utilices siempre que sea explícitamente un editor de texto plano como Atom o Notepad++. Recuerda: ya no estamos atados a la herramienta, se puede cambiar de editor en cualquier momento.
- **Terminal de línea de comandos.** Trabajar en la “línea de comandos” es lo mismo que escribir comandos en la terminal. En Mac sólo tienes que utilizar tu *Finder* para acceder a “Terminal”. En Windows utiliza *PowerShell*. Es probable

que los usuarios de Linux ya estén familiarizados con sus terminales. A continuación, vamos a cubrir los conceptos más básicos de cómo encontrar y utilizar la línea de comandos.

- **Pandoc.** Las instrucciones detalladas de instalación específica para cada plataforma están disponibles en el [sitio web de Pandoc](#). *Para este tutorial es crucial que instales Pandoc en tu ordenador*, así que asegúrate de invertir tiempo navegando por las instrucciones. Pandoc fue creado y es mantenido por John MacFarlane, profesor de Filosofía en la Universidad de California en Berkeley. Esto es *humanidades digitales* en su mejor expresión y servirá como el motor de nuestro flujo de trabajo. Con Pandoc serás capaz de compilar el texto y la bibliografía de tu trabajo en documentos con un formato flexible y atractivo. Una vez que hayas seguido las instrucciones de instalación, verifica su instalación escribiendo en la línea de comandos de tu máquina “`pandoc -version`”. Asumimos que por lo menos tienes la versión 1.12.3 publicada en enero de 2014.

Recomendamos que instales los dos siguientes programas de aplicación, aunque no son un requisito indispensable para completar este tutorial.

- **Zotero o Endnote.** El *software* para referencias bibliográficas como Zotero y Endnote son herramientas indispensables para organizar y formar citaciones en un trabajo de investigación. Estos programas pueden exportar tus bibliotecas como un archivo BibTeX (sobre el que aprenderás inmediatamente en el caso 2, más abajo). Este archivo, que es en sí mismo un documento de texto con el formato de todas tus referencias bibliográficas, te permitirá citar publicaciones rápida y fácilmente utilizando `@tags`. Cabe señalar que también es posible escribir todas tus referencias bibliográficas a mano usando [nuestra bibliografía](#) como plantilla.
- **LaTeX.** Las instrucciones detalladas para la instalación específica en cada plataforma están disponibles en el [sitio web de Pandoc](#). A pesar de que este tutorial no cubre LaTeX, éste es utilizado por Pandoc para la creación de PDF. Los usuarios suelen convertir en LaTeX directamente para tener un control más minucioso de la composición tipográfica de los .pdf. Los principiantes pueden considerar saltarse este paso. De lo contrario, escribe en tu terminal `latex -v` para ver si LaTeX se ha instalado correctamente (obtendrás un error si así no fuera y algo de información sobre la versión si fue exitosa).

## Bases de Markdown

Markdown es una convención para estructurar tus documentos en texto plano de una manera semántica. La idea es identificar estructuras lógicas en tu documento (un título, una sección, subsecciones, notas al pie, etc.), marcarlas con algunos caracteres distintivos y luego “compilar” el texto resultante con un intérprete de composición tipográfica que

dará forma al documento en consonancia con el estilo especificado.

Las convenciones para Markdown están disponibles en varios tipos o “*flavors*”, diseñados para su uso en contextos particulares como blogs, *wikis* o repositorios de código. El *flavor* de Markdown utilizado por Pandoc está orientado para un uso académico. Sus convenciones están descritas en la página de [Pandoc’s Markdown](#). Estas convenciones incluyen el “YAML” block, que contiene una serie de metadatos muy útiles. <sup>5</sup>

Vamos a crear ahora un documento simple en Markdown. Abre tu editor de texto plano seleccionado y escribe algo que debe tener un aspecto como el siguiente:

```
---
title: Flujo de trabajo en texto plano
author: Gabriel García
date: 20 de enero de 2014
fontfamily: times
---
```

El Markdown “Pandoc-flavored” almacena cada uno de los valores anteriores y los “imprime” en la ubicación apropiada de tu documento de salida una vez que está listo para la composición tipográfica. Más adelante aprenderemos a incluir campos más potentes en YAML. Por ahora, vamos a suponer que estamos escribiendo un documento compuesto por tres secciones, cada una subdividida en dos. Hay que dejar una línea en blanco después de los tres últimos guiones del bloque YAML para que puedas pegar lo que sigue:

```
# Sección 1

## Subsección 1.1

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim
ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut
aliquip ex ea commodo consequat.

El siguiente párrafo debe empezar como éste, sin sangría:

## Subsección 1.2
```

```
Sed ut perspiciatis unde omnis iste natus error sit voluptatem  
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa  
quae ab illo inventore veritatis et quasi architecto beatae vitae  
dicta sunt explicabo.
```

```
# Sección 2
```

```
## Subsección 2.1
```

Sigue adelante e introduce cualquier texto de relleno. Los espacios vacíos tienen significado en Markdown por lo que no debes poner sangría en los párrafos pero sí es importante que separes los párrafos con una línea en blanco. Las líneas en blanco también deben preceder a los encabezados de sección.

Puedes añadir asteriscos para dar énfasis a las palabras con negritas o cursivas de esta manera: *\*cursivas\** y **\*\*negritas\*\***. También hay que añadir a nuestro texto un enlace y una nota a pie de página para cubrir los requisitos de un texto promedio. Escribe:

```
Una oración que requiere una cita.[^1]
```

```
[^1]: ¡Ésta es mi primer nota a pie de página! Y un [enlace]  
(https://www.eff.org/).
```

Cuando el texto del enlace y la dirección del mismo son iguales es más rápido escribir:

```
<https://www.eff.org/>, en vez de [https://www.eff.org/]  
(https://www.eff.org/).
```

Vamos a guardar nuestro archivo antes de ir más lejos. Haz una carpeta para albergar este proyecto. Es probable que tengas un sistema de organización de tus documentos, proyectos, ilustraciones y bibliografías, pero a menudo tu documento, tus proyectos, tus ilustraciones y bibliografías se encuentran en diferentes carpetas, lo que los hace difíciles de encontrar. Nuestro objetivo es crear una carpeta única para cada proyecto con todos los materiales relevantes incluidos en ella. La regla general es “un proyecto, un texto, una carpeta”. Denomina a tu archivo algo así como “principal.md”, donde “md” significa que es un archivo Markdown.

Una vez que has guardado el archivo, vamos a añadir una imagen. Copia una imagen pequeña a la carpeta y añade lo siguiente en alguna parte del cuerpo de texto: `![una imagen](tu_imagen.jpg)`.

En este punto tu archivo `principal.md` debe verse como sigue



```
---
title: Flujo de trabajo en texto plano
author: Gabriel García
date: 20 de enero de 2014
---

# Sección 1

## Subsección 1.1

Lorem *ipsum* dolor sit amet, **consectetur** adipisicing elit, sed
do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut
enim ad minim veniam, quis nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo consequat.

El siguiente párrafo debe empezar como este, sin sangría:

## Subsección 1.2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem
accusantium doloremque laudantium, totam rem aperiam, eaque ipsa
quae ab illo inventore veritatis et quasi architecto beatae vitae
dicta sunt explicabo.

# Sección 2

## Subsección 2.1
![una imagen](tu_imagen.jpg)

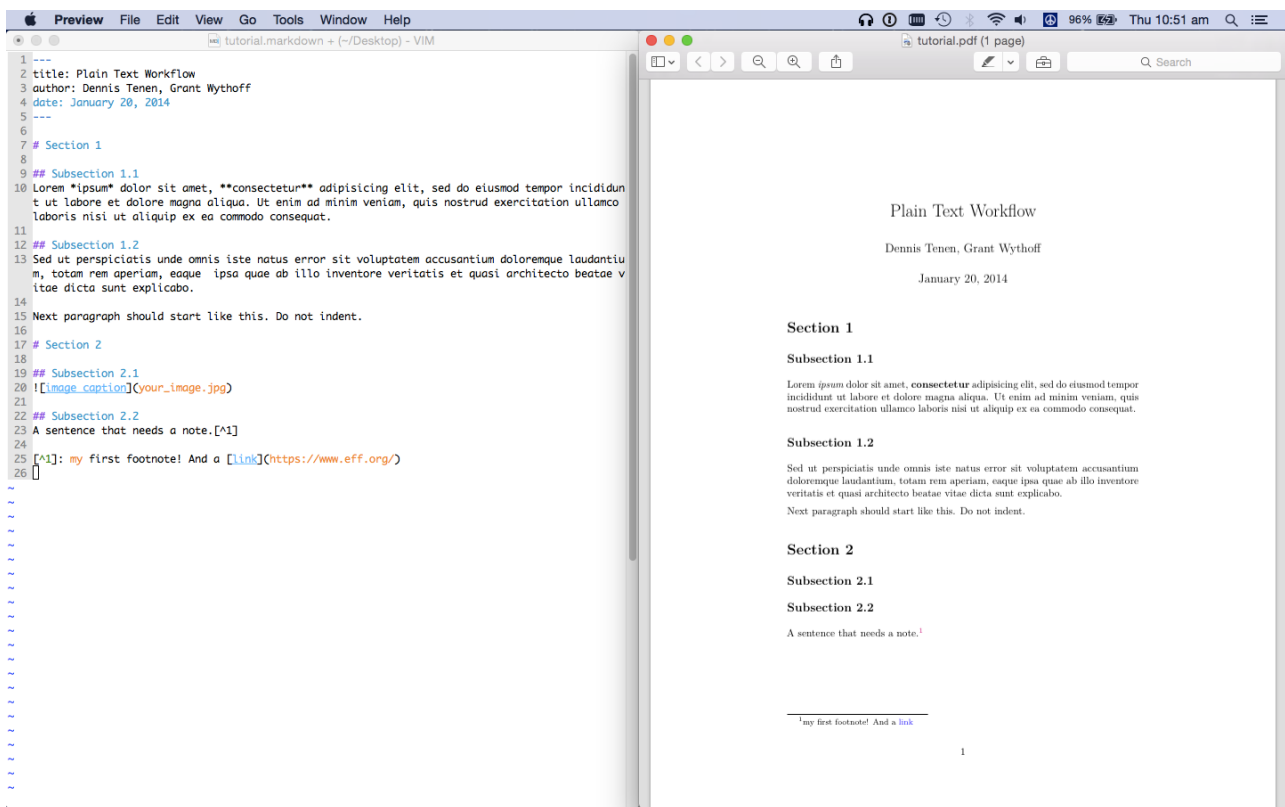
## Subsección 2.2.

Una oración que requiere una cita.[^1]

[^1]: ¡Ésta es mi primer nota a pie de página! Y un [enlace]
(https://www.eff.org/).
```

Y como veremos en breve, este archivo de texto plano se puede representar como un muy buen PDF.





## Captura de pantalla de un PDF interpretado por Pandoc

Si quieres tener una idea de cómo serán interpretado en un fomato HTML este tipo de marcado, prueba este [sitio de prueba en línea](#) y juega con varios tipos de sintaxis. Recuerda que ciertos elementos del *Pandoc-flavored markdown* (como el bloque de título o las notas al pie) no funcionan en esta versión web ya que solamente acepta lo básico.

En este punto, deberás ocupar algún tiempo explorando algunas de las características de Markdown como las citas de texto (referidas con el símbolo `>`), los listados que empiezan con `*` o `-`, los saltos de línea literales que empiezan con `||` (útiles para poesía), las tablas y algunas otras funciones señaladas en la página sobre Markdown de Pandoc.

Presta particular atención a los espacios en blanco y al flujo de los párrafos. La documentación lo explica sucintamente cuando define un párrafo como “una o más líneas de texto seguidas por una o más líneas en blanco.” Consiedra que “las nuevas líneas son tratadas como espacios” y que “si necesitas un salto de línea elocuente, utiliza dos o más espacios en blanco al final de la línea.” La mejor manera de entender lo que significa es experimentar libremente. Utiliza el modo de vista previa de tu editor o solamente ejecuta Pandoc para ver los resultados de tus experimentos.

Pero sobre todo, evita la necesidad de formatear. Recuerda que estás identificando unidades semánticas: secciones, subsecciones, énfasis, notas al pie y figuras. Incluso `*cursivas*` y `**negritas**` en Markdown no son en realidad marcas de formato, sino que indican un nivel diferente de énfasis. La aplicación del formato sucederá después, una

vez que conozcas el momento del proceso en el que hay que hacerlo y los requerimientos de la publicación.

Existen programas que te permiten obtener una vista previa en vivo de la salida de markdown al tiempo que editas tu archivo de texto plano y que detallaremos más adelante en la sección de Recursos útiles. Algunos de ellos soportan notas a pie, figuras, incluso bibliografías. Sin embargo, para sacar provecho al máximo de Pandoc, te recomendamos que te quedes con lo más sencillo: archivos de texto plano almacenados localmente en tu computadora.

## Estar en contacto con la terminal de la máquina

Antes de que podamos publicar nuestro archivo `principal.md` en otros formatos, necesitamos un poco de orientación para trabajar en la interfaz de línea de comandos utilizando la aplicación de la terminal de nuestro ordenador, que es la única manera (y la mejor) de utilizar Pandoc.

La interfaz de línea de comandos es un sitio amigable una vez que te acostumbras a ella. Si ya te es familiar la utilización de línea de comandos, puedes saltarte esta sección. Para el resto de lectores, es importante entender que al usar la terminal directamente podrán acceder a una amplia gama de herramientas de investigación poderosas que de otra manera es imposible, y será la base para trabajos más avanzados. Para el propósito de este tutorial necesitas aprender solamente unos cuantos comandos muy sencillos.

Primero, abre una nueva ventana de línea de comandos. Si utilizas una macOS, abre Terminal en el directorio `Aplicaciones/Utilidades`. En Windows utilizarás PowerShell. En Windows 7 o posterior, recomendamos que utilices el “powershell” o, para una solución más robusta, instala el subsistema de Windows para Linux y utiliza la terminal que viene con tu distribución Linux favorita. Para una excelente introducción a la línea de comando, consulta “[Introduction to the Bash Command Line](#)” por Ian Milligan y James Baker.

En el terminal debe ver una ventana de texto y un puntero (*prompt*) que puede verse más o menos como esto: `nombre-del-ordenador:~nombre-de-usuario$`. La tilde indica que estás en el directorio de usuario, y de hecho puedes escribir `$ cd ~` en cualquier punto para regresar a tu directorio de usuario. No escribas el símbolo de moneda pues solamente indica el puntero de tu terminal, solicitándote que escribas algo en la terminal (como opuesto a que escribas algo en un documento); recuerda introducir Enter después de escribir cada comando.

Es muy común que tu carpeta “Documents” esté localizada en este directorio. Escribe `$ pwd` (= “*print working directory*”) y oprime Enter para mostrar el nombre del directorio actual. Utiliza `$ pwd` cada vez que sientas que te has extraviado.

El comando `$ ls` (= *list*), simplemente enlista los archivos en el directorio actual. Finalmente, puedes usar `$ cd` (= *change directory*) como `$ cd NOMBRE_DE_LA_CARPETA` (en donde `NOMBRE_DE_LA_CARPETA` indica la carpeta en la que quieres navegar). Puedes utilizar `$ cd ..` para moverte automáticamente un nivel arriba en la estructura del directorio (el directorio principal de la carpeta en la que te encuentras actualmente). Una vez que has empezado a escribir el nombre de la carpeta, utiliza la tecla de tabulador para autocompletar el nombre -lo cual es particularmente útil para carpetas con nombre muy largos o nombres de carpetas que contienen espacios en blanco.<sup>6</sup>

Estos tres comandos de terminal: `pwd`, `ls` y `cd` es todo lo que necesitas en este tutorial. Practica con ellos la navegación por las carpetas de tus documentos por unos minutos y mientras piensa en la manera en la que has organizado tus archivos. Si lo deseas, sigue lo que haces ayudándote de tu organizador de archivos de la interfaz gráfica de usuario (Finder) para poder orientarte.

## Usar Pandoc para convertir Markdown a un documento de MS Word

¡Ya estamos listos para la composición tipográfica! Abre la ventana de tu terminal, utiliza `$ pwd` y `$ cd NOMBRE_DE_LA_CARPETA` para navegar hasta la carpeta en la que se encuentra tu proyecto. Una vez que estés ahí escribe `$ ls` en la terminal para enlistar los archivos. Si ves tu archivo `.md` y tus imágenes te encuentras en el lugar correcto. Para convertir `.md` a `.docx` escribe:

```
$ pandoc principal.md -o principal.docx
```

Abre el archivo con MS Word para contejar tus resultados. Si utilizas Open- o Libre Office puedes ejecutar:

```
$ pandoc principal.md -o principal.odt
```

Si la línea de comandos es una novedad para ti, imagina que lees los comandos anteriores y que dicen algo como: “Pandoc: crea un archivo MS Word a partir de mi archivo Markdown.” La partícula `-o` es una “bandera” que en este caso dice algo como “en vez de que explícitamente tenga yo que indicarte la fuente y el objeto de los formatos de archivos, adivina con sólo ver la extensión de los archivos.” Hay muchas opciones disponibles en Pandoc a través de estas “banderas”. Puedes ver la lista completa en el [sitio web de Pandoc](#), o escribiendo en la terminal:

```
$ man pandoc
```

Intenta ejecutar el comando:

```
$ pandoc principal.md -o proyecto.html
```

Ahora navega de nuevo por el directorio de tu proyecto. ¿Puedes decir qué sucedió?

Los usuarios más avanzados que tienen instalado LaTeX querrán experimentar con la conversión de markdown a un archivo `.txt` o a un archivo `.pdf` formateado especialmente. Una vez que se ha instalado LaTeX se pueden crear archivos PDF bellamente formados utilizando la misma estructura de comandos:

```
$ pandoc -o principal.pdf principal.md
```

Asegúrate que tu editor de texto soporte el formato de codificación UTF-8. Cuando utilices LaTeX para convertir al formato `.pdf`, en vez del atributo `fontfamily` en YAML para cambiar la fuente, especifica el atributo `mainfont` para producir algo como esto:

---

```
title: Flujo de trabajo en texto plano
author: Dennis Tenen, Grant wythoff
date: 20 de enero de 2014
mainfont: times    ---
```

## Trabajar con bibliografías

En esta sección agregaremos una bibliografía a nuestro documento y después la convertiremos de un formato estilo Chicago a un formato estilo MLA.

Si no estás usando un gestor de referencias bibliográficas como Endnote o Zotero, deberías comenzar a hacerlo inmediatamente. Nosotros preferimos Zotero porque, al igual que Pandoc, fue creado por la comunidad académica y, al igual que otros proyectos de código abierto, es distribuido con una Licencia Pública General de GNU. Mucho más importante para nosotros es que tu gestor de referencias tenga la habilidad de generar bibliografías en formato de texto plano para estar en consonancia con nuestro principio de “todo en texto plano”. Prosigue y abre el gestor de referencias de tu elección y añade algunas entradas de ejemplo. Cuando hayas terminado, encuentra la opción para exportar tu bibliografía en formato BibTeX (`.bib`). Guarda tu archivo `.bib` en el directorio de tu proyecto y dale un nombre razonable como `proyecto.bib`.

La idea general es mantener tus fuentes organizadas en una base de datos bibliográfica centralizada mientras vas generando archivos `.bib` específicos y más pequeños que serán almacenados en el mismo directorio de tu proyecto. Prosigue y abre tu archivo `.bib` con el editor de texto plano que hayas elegido.<sup>7</sup>

Tu archivo `.bib` deberá contener múltiples entradas que se ven más o menos así:

```
@article{fyfe_digital_2011,  
  title = {Digital Pedagogy Unplugged},  
  volume = {5},  
  url =  
{http://digitalhumanities.org/dhq/vol/5/3/000106/000106.html},  
  number = {3},  
  urldate = {2013-09-28},  
  author = {Fyfe, Paul},  
  year = {2011},  
  file = {fyfe_digital_pedagogy_unplugged_2011.pdf}  
}
```

Rara vez tendrás que editar esto a mano (aunque puedes hacerlo). En la mayoría de los casos, simplemente exportas el archivo `.bib` de Zotero o de un gestor de referencias similar. Tomate un tiempo para orientarte en esto. Cada entrada consiste en un tipo de documento, “artículo” (*article*) en nuestro caso, un identificador único (`fyfe_digital_2011`) y los metadatos relevantes de título (*title*), autor (*author*), etc. Lo que más nos interesa es el identificador (ID) único que sigue inmediatamente al símbolo de llave ( `{ }` ) en la primera línea de cada entrada. El ID único es lo que nos permite conectar la bibliografía con el documento principal. Deja este archivo abierto por ahora y regresa a tu archivo `principal.md`.

Edita la nota a pie de página en la primera línea de tu archivo `principal.md` para que se vea de una forma parecida a los siguientes ejemplos en los cuales `@nombre_título_fecha` puede ser reemplazado por uno de los ID únicos de tu archivo `proyecto.bib`.

- Una referencia bibliográfica formateada como ésta se traducirá apropiadamente tanto en un estilo de citación en texto -como en nota a pie- `[@nombre_título_fecha, 67]`.<sup>8</sup>
- "Para citas entrecomilladas, pon la coma afuera de los signos de las comillas" `[@nombre_título_fecha, 67]`.

Una vez que ejecutes el markdown a través de Pandoc, “@fyfe\_digital\_2011” se ampliará a una citación completa en el estilo que hayas seleccionado. Puedes usar la sintaxis `@citacion` de cualquier manera que veas que encaja: dentro de las líneas de tu texto o en las notas a pie. Para generar una bibliografía simplemente incluye una sección llamada `# Bibliografía` al final del documento.

Ahora, vayamos de nuevo a nuestro bloque de metadatos en el encabezado de tu documento `.md`, y especifica el archivo de bibliografía que deberá utilizarse, algo como:

```
---
title: Flujo de trabajo en texto plano
author: Gabriel García
date: 20 de enero de 2014
bibliography: proyecto.bib
---
```

Esto le dice a Pandoc que busque tu bibliografía en el archivo `proyecto.bib` dentro del mismo directorio de tu archivo `principal.md`. Veamos si esto trabaja. Guarda tu archivo, ve a la ventana de terminal y ejecuta:

```
$ pandoc principal.md --filter pandoc-citeproc -o principal.docx
```

El filtro “pandoc-citeproc” compila todas tus etiquetas de citas. El resultado debe ser un archivo de MS Word formateado decentemente. Si tienes instalado LaTeX, conviértelo a `.pdf` utilizando la misma sintaxis para mejores resultados. No te preocupes si las cosas no aparecen exactamente de la manera que tú quisieras -recuerda que vas a afinar el formato de todo una vez y más tarde, lo más cerca posible del momento de la publicación. Por ahora solamente estamos creando borradores basados en valores por defecto.

## Cambiar los estilos de citación

El estilo de citación por defecto en Pandoc es el de Chicago Autor-fecha. Podemos especificar un estilo diferente utilizando una hoja de estilo escrita en “lenguaje de estilo de citación” (CSL por *citation style language*, otra convención en texto plano utilizada para describir estilos de citas) y que es designado por la extensión de archivo `.cs1`.

Afortunadamente, el proyecto CSL mantiene un repositorio de estilos de citaciones comunes, algunas incluso ajustadas a ciertas revistas en específico. Visita <http://editor.citationstyles.org/about/> para encontrar el archivo `.cs1` para el estilo Modern Language Association (MLA), descarga el archivo `modern-language-association.cs1` y guárdalo en la carpeta de tu proyecto como `mla.cs1`. Ahora, necesitamos indicarle a Pandoc que

utilice la hoja de estilo de MLA en vez de la de Chicago que tiene por defecto. Haremos esto actualizando el encabezado o bloque YAML:

```
---
title: Flujo de trabajo en texto plano
author: Gabriel García
date: 20 de enero de 2014
bibliography: proyecto.bib
csl: mla.csl
---
```

Después simplemente utiliza la funcionalidad de Pandoc para transformar tu archivo de markdown a tu formato objetivo (.pdf o .docx):

```
$ pandoc principal.md --filter pandoc-citeproc -o principal.pdf
```

## Resumen

Ahora debes ser capaz de escribir artículos en Markdown, crear borradores en varios formatos, añadir bibliografías y cambiar estilos de citación de manera sencilla. Un vistazo final al directorio de tu proyecto te mostrará un número de archivos de origen de datos: tu archivo `principal.md`, el archivo `proyecto.bib`, el archivo `mla.csl`, y algunas imágenes. Además de los archivos de origen, deberías ver algunos archivos de salida que creamos durante el tutorial: `principal.docx` o `principal.pdf`. Tu carpeta debe verse más o menos de esta manera.

```
tutorial-Pandoc/
  principal.md
  proyecto.bib
  mla.csl
  image.jpg
  principal.docx
```

Trata tus archivos de origen como versiones autorizadas de tu texto y los archivos de salida como impresiones desechables que puedes generar fácilmente y sobre la marcha con Pandoc. Todas las revisiones deben ir dentro del archivo `principal.md`. El archivo `principal.docx` está ahí para la última etapa de limpieza y formato. Por ejemplo, si la revista requiere manuscritos a doble espacio, puedes darle el doble espacio rápidamente en Open Office o Microsoft Word. Pero no gastes demasiado tiempo formando. Recuerda



que el manuscrito debe ir despojado de todo cuando va a imprenta. El tiempo dedicado a formar cosas innecesarias puede aprovecharse mejor en pulir la prosa de tu borrador.

## Recursos útiles

En caso de meterte en problemas no hay un mejor lugar para empezar a buscar soluciones que el [sitio web de Pandoc](#) de John MacFarlane y la [lista de correos](#) afiliada (en inglés). Al menos en dos sitios de tipo “Pregunta y respuesta” puedes encontrar respuestas a preguntas sobre Pandoc: [Stack Overflow](#) y [Digital Humanities Q&A](#). Puedes hacer preguntas en vivo en Freenode IRC, #Pandoc channel, frecuentado por un amistoso grupo de asiduos. A medida que aprendas más acerca de Pandoc, puedes explorar una de sus particularidades más poderosa: [filtros](#).

Aunque te sugerimos comenzar con un simple editor de texto plano, hay muchas más alternativas (más de 70, de acuerdo con [esta entrada de blog](#) a MS Word para trabajar específicamente con Markdown, disponibles en línea y a menudo sin costo. Para las autónomas nos gustan [Mou](#), [Write Monkey](#), y [Sublime Text](#). Varias plataformas web que han surgido recientemente proporcionan interfaces gráficas adecuadas para desarrollar una escritura colaborativa con seguimiento de cambios en las versiones utilizando Markdown. Éstas incluyen: [prose.io](#), [Authorea](#), [Penflip](#), [Draft](#), y [StackEdit](#).

Pero el ecosistema no está limitado sólo a editores. [Gitit](#) e [Ikiwiki](#) soportan escritura en Markdown utilizando Pandoc como compilador. A esta lista se puede agregar una serie de herramientas que generan páginas web estáticas de manera rápida: [Yst](#), [Jekyll](#), [Hakyll](#) y [bash shell script](#) por el historiador Caleb McDaniel.

Finalmente, se están creando plataformas de publicación enteras basadas en el uso de Markdown. La plataforma de mercado [Leanpub](#) puede ser una alternativa interesante al modelo tradicional de publicación y nosotros mismos estamos experimentando con el diseño de una revista académica en GitHub y [readthedocs.org](#) (herramientas que suelen utilizarse para técnicas de documentación).

1. ¡No te preocupes si no entiendes aún esta terminología! ↩
2. [GitHub](#). Utiliza la opción “raw” cuando lo veas en GitHub para observar la fuente de Markdown. Los autores queremos agradecer a Alex Gil y sus colegas del Columbia’s Digital Humanities Center, y a los participantes de openLab en el Studio de la Biblioteca Butler por probar el código de este tutorial en diversas plataformas. ↩
3. Véase la excelente discusión sobre este tema, por Charlie Stross, en [Why Microsoft Word Must Die](#). ↩
4. Considera que la extensión `.bib` debe estar “vinculada” a Zotero en tu sistema operativo. Esto significa que si haces doble click en un archivo `.bib`, es probable

que Zotero intente abrir el archivo mientras que nosotros queremos abrirlo con un editor de texto. Es posible que en el futuro quieras asociar la extensión `.bib` a tu editor de texto. ↩

5. Ten en cuenta que a menudo el YAML replica algo, aunque no todo, de la funcionalidad (bandera) de la línea de comando. Por ejemplo, los estilos de fuentes pueden pasarse a Pandoc en la forma de `pandoc principal.md --mainfont=times -o target.pdf`. Sin embargo, preferimos utilizar las opciones de encabezado YAML siempre cuando sea posible, pues hace la funcionalidad de nuestra línea de comandos más fácil de escribir y recordar. Utilizando una herramienta de control de cambios como Git preservará tus cambios al YAML, mientras lo que escribes en la terminal es más efímero. Consulta la sección de plantillas en el manual de Pandoc (`man pandoc`) para ver la lista de variables YAML disponibles. ↩
6. No hay buenas soluciones para traducir a MS Word desde LaTeX. ↩
7. Es una buena idea crearse el hábito de no usar espacios en el nombre de una carpeta o un archivo. Los guiones y guiones bajos en vez de espacios en los nombres de archivo aseguran una compatibilidad perdurable entre plataformas cruzadas. ↩
8. Gracias a [@njbart](#) por la corrección. En respuesta a nuestra sugerencia original: `Alguna frase que necesita citación.^[@fyfe_digital_2011 argues that too.]` él escribe: “This is not recommended since it keeps you from switching easily between footnote and author-date styles. Better use the [corrected] (no circumflex, no final period inside the square braces, and the final punctuation of the text sentence after the square braces; with footnote styles, pandoc automatically adjusts the position of the final punctuation).”