

Solución no propietaria para enlaces entre notas de un Zettelkasten digital

Eduardo Riesco

abril 2022

Resumen.

Existe una amplia variedad de aplicaciones software para gestión de conocimiento (Personal Knowledge Management), siendo las más interesantes aquellas que nos permiten interrelacionar ideas, normalmente escritas en forma de notas. Los datos manejados por estas aplicaciones los constituyen las propias notas y sus metadatos. Como usuarios, desearíamos mantener esos datos en un formato exportable y estándar que evite el “vendor lock-in” con la aplicación elegida permitiéndonos mudanzas futuras si así lo deseamos. Las notas en sí mismas no suelen constituir un problema ya que normalmente se almacenan en ficheros en algún lenguaje estándar de marcado ligero como Markdown. El reto lo constituyen los metadatos ya que, a falta de un estándar, cada aplicación usa su propio formato y forma de almacenamiento. Acotamos el problema cuando entendemos que de esos metadatos son los enlaces entre notas lo realmente relevante. La siguiente propuesta explora el uso de #hashtags para el enlace entre notas independientemente de la aplicación utilizada, buscando una solución centrada en los datos e independiente del software.

El Zettelkasten de Luhmann

Quizá el mayor y más influyente análisis sobre las ventajas de enlazar nuestras notas entre sí para generar conocimiento fue el realizado por el sociólogo alemán Niklas Luhmann¹ a mediados del siglo XX.

El método Zettelkasten² de Luhmann está basado en la interconexión de ideas escritas en notas. Esta red de conocimiento se convierte así en una herramienta

¹<https://niklas-luhmann-archiv.de/>

²<https://zettelkasten.de/posts/overview/>

para pensar y escribir. Es la *conexión* entre ideas lo que le aporta esta característica y no la simple *colección* de notas. El valor reside en la red que forman. Posteriores autores han denominado a esta red *segundo cerebro* aunque Luhmann prefería referirse a ella como su *compañero de comunicación*.³

En una implementación clásica de Zettelkasten se usan tarjetas físicas para escribir las notas⁴, dándole un **identificador único** a cada tarjeta. La conexión entre ellas se establece escribiendo en cada una los identificadores de las tarjetas relacionadas.

El punto de entrada a las notas lo constituye el **registro**⁵, formado por una colección de etiquetas (*keywords*)⁶ asociadas a notas que se consideran el inicio de un argumento o tema concreto.

En un **Zettelkasten digital** las notas suelen escribirse en un editor de texto plano usando algún lenguaje de marcado ligero como **Markdown**⁷ o similar. Cada nota se almacenará después en un archivo separado.

En esta solución digital, la **conexión entre notas** puede abordarse de diferentes maneras. Lo habitual es usar aplicaciones específicas para gestión de Zettelkasten que se encargan de mantener algún tipo de registro con los enlaces entre notas. Esto une indefectiblemente nuestro Zettelkasten a una aplicación concreta, haciéndolo dependiente de ella (*vendor lock-in*)⁸. Si en un futuro deseáramos reemplazar esta aplicación sería difícil recuperar los enlaces entre las notas. En función del método usado por la aplicación para codificar las relaciones, el “rescate” de nuestro Zettelkasten para usarlo en otra aplicación diferente puede llegar a ser complicado.

No ocurre lo mismo con el texto de las notas, escrito en un lenguaje de marcado legible usando cualquier otro editor. Almacenarlo en archivos que utilizan un formato estándar nos proporciona esta independencia entre los datos y la aplicación.

Wikilinks como estado del arte

Las aplicaciones para gestión de notas suelen usar `[[wikilinks]]` para enlazar notas. El concepto de *wikilink*⁹ permite, mediante una sintaxis sencilla, enlazar unidireccionalmente nodos de datos pertenecientes a una misma aplicación. El ejemplo más habitual son los artículos de una wiki.

³<https://luhmann.surge.sh/communicating-with-slip-boxes>

⁴<https://niklas-luhmann-archiv.de/bestand/zettelkasten/suche>

⁵https://niklas-luhmann-archiv.de/bestand/zettelkasten/zettel/ZK_2_SW1_001_V

⁶https://en.wikipedia.org/wiki/Index_term

⁷<https://daringfireball.net/projects/markdown/>

⁸https://es.wikipedia.org/wiki/Dependencia_del_proveedor

⁹<https://en.wikipedia.org/wiki/Hyperlink#Wikis>

Para conseguir que el enlace entre nodos sea bidireccional es necesario crear un *backlink*¹⁰, es decir, un enlace desde el nodo destino al nodo original. La aplicación suele encargarse de crearlos y gestionarlos automáticamente. En la Wikipedia, por ejemplo, los *backlinks* de una página se muestran al pulsar sobre “Lo que enlaza aquí”.

Los *wikilinks* no forman parte del estándar Markdown aunque son casi un estándar de facto y se incluyen en muchas de sus extensiones. Aun así, los editores de texto no suelen reconocerlos y solo es posible usarlos para navegar entre notas usando búsqueda de texto global.

Requieren también identificadores únicos para los nodos a enlazar. Normalmente este identificador será el nombre del fichero o nodo que almacena la nota, perdiéndose el enlace si se modifica el nombre. A veces la aplicación se encarga de modificar los enlaces si detecta cambios en el nombre del nodo. En otros casos el identificador se sitúa en alguna parte de la nota con una sintaxis especial tipo clave-valor, independizándolo del nombre del fichero o se almacenan en una base de datos ligera tipo SQLite.

Hiperenlaces como primera aproximación

Idealmente sería deseable encontrar una codificación para los enlaces entre notas de alguna forma estandarizada independiente de la aplicación.

Los lenguajes de marcado ligero en los que suelen escribirse las notas incluyen formas de expresar **hiperenlaces**¹¹ a webs o a archivos locales. Esa característica podría aprovecharse para enlazar notas situadas en diferentes archivos. En ese caso se estaría usando el **nombre del archivo como identificador único** de la nota. Este enfoque tiene varios **inconvenientes**:

- Si se modifica el nombre o la localización del archivo de destino será necesario buscar y modificar todos los hiperenlaces que apunten a él.
- Un hiperenlace común es unidireccional mientras que las notas en un Zettelkasten deben estar relacionadas mediante enlaces bidireccionales.
- Los editores de texto normalmente no permiten pulsar en un enlace expresado de esta manera para abrir otro archivo en el mismo editor.

Estas soluciones no parecen las más convenientes y se hace necesario buscar otra forma de enlazar notas referenciándolas entre sí.

¹⁰<https://en.wikipedia.org/wiki/Backlink>

¹¹<https://es.wikipedia.org/wiki/Hiperenlace>

Hashtags como solución

En virtud del análisis anterior se propone **usar etiquetas (tags) para enlazar notas**. Es decir, todas las notas que contengan la misma etiqueta se considerarán enlazadas. De hecho, las aplicaciones específicas de gestión de notas ya suelen incorporar la posibilidad de añadir etiquetas. Esto es, en efecto, otra manera usada por estas aplicaciones para relacionar notas que no es demasiado diferente a los enlaces directos. Al usar solamente etiquetas estamos unificando la forma de enlazar notas mediante una **sintaxis única**.

Los lenguajes de marcado ligero no contemplan ningún estándar para etiquetar documentos. Se usará un **convenio** bastante común según el cual las etiquetas se crean usando el símbolo `#` seguido de cualquier carácter diferente a un espacio. P. ej: `#tag1`. Este convenio no interfiere con ningún elemento de marcado contemplado en el estándar Markdown. Las etiquetas expresadas de esta forma suelen denominarse *hashtags*¹² y son comunes en redes sociales con un significado similar.

Las etiquetas podrían situarse en cualquier parte del texto, por ejemplo en el *front-matter*¹³ al inicio del documento:

```
---
title: "Habits"
author: John Doe
date: March 22, 2005
tags: #tag1 #tag2 #tag3
---
```

intercalando entre el texto el **#concepto** que queremos enlazar o en una línea final aislada para enlazar la nota en su totalidad:

```
#tag1 #tag2 #tag3
```

El uso de etiquetas tiene varias **ventajas** frente a otros métodos:

- Evita requerir identificadores únicos para las notas. Una etiqueta constituye un **identificador no necesariamente único** que permite referenciar una nota para ser enlazada.
- Una etiqueta forma de manera intrínseca una unión **bidireccional** entre los documentos que la contengan.
- Un enlace establece una relación 1:1 entre dos notas. Esta misma relación se puede conseguir asignando la misma etiqueta a esas dos notas.
- Una etiqueta también puede establecer un **conjunto** de notas. Esta idea permite crear subconjuntos de temas específicos dentro de conjuntos más generales o visualizar más claramente las intersecciones entre ideas.

¹²<https://es.wikipedia.org/wiki/Hashtag>

¹³https://en.wikipedia.org/wiki/Book_design#Front_matter

- Las etiquetas añaden **contenido semántico** a los enlaces y conjuntos que forman al asignarles un nombre.
- Aunque solemos asociar etiquetas a conjuntos, es posible usarlas para enlazar **razonamientos** tipo $A \rightarrow B \rightarrow C \rightarrow D$ mediante `#tag1 → #tag1,#tag2 → #tag2,#tag3 → #tag3`
- En general, en cualquier editor o aplicación, para **mostrar un conjunto de notas enlazadas** se pueden usar las herramientas estándar de búsqueda global de texto. Simplemente buscando el nombre de la etiqueta obtendremos el conjunto de notas enlazadas por ella. Esto evita codificar extensiones específicas.

Aunque según este último punto es posible navegar entre notas enlazadas usando un editor de texto común, también es cierto que no es posible saltar entre notas pulsando sobre las etiquetas y que no todos los editores tienen búsqueda global en varios ficheros.

Algunas aplicaciones para Zettelkasten como *Zettel*, *Obsidian*, *Roam Research*, *logseq*, etc. usan también *hashtags* como convenio para insertar etiquetas por lo que se podrán aprovechar las opciones de búsqueda y navegación que proporcionan. Normalmente también es posible moverse entre notas pulsando sobre las etiquetas.

Crear enlaces mediante etiquetas de esta manera permite gestionar nuestro repositorio de notas con cualquier aplicación a costa de perder la funcionalidad específica que pueda tener para enlazar notas. Se perderán, por ejemplo, las opciones relativas a los gráficos de nodos y vértices para notas enlazadas aunque es posible utilizar alguna herramienta externa que analice los ficheros y genere estos gráficos.

En resumen, mantendremos el control de nuestros datos siempre que usemos *hashtags* como mecanismo para enlazar notas, absteniéndonos de utilizar las soluciones propietarias que nos ofrecen las diferentes aplicaciones.

Conclusión

El enlace entre notas es la característica definitoria de un Zettelkasten y son precisamente las soluciones propietarias para estos enlaces lo que nos hace depender de una u otra aplicación específica. Empleamos mucho tiempo eligiendo cuidadosamente nuestro gestor de notas sabiendo que es una elección sin una vuelta atrás sencilla y porque no reparamos en que es la gestión de los enlaces lo único realmente importante. Si mantenemos el control sobre ellos podemos cambiar entre aplicaciones en función de nuestra preferencia por unas u otras características secundarias o incluso usar varias aplicaciones al mismo tiempo sobre un único repositorio de notas. Los gestores de notas no deberían ser más que diferentes *visores* de unos datos sobre los que nunca perdemos el control.

Luhmann necesitaba una manera de enlazar sus notas y encontró una solución

con las herramientas disponibles en la época. Actualmente se intenta replicar casi exactamente esa misma solución con herramientas modernas sin pensar que si Luhmann hubiera tenido acceso a soluciones digitales probablemente habría llegado a otras conclusiones y habría estructurado su Zettelkasten de manera diferente. Él utilizó identificadores únicos y enlaces escritos a mano porque no existen los buscadores de texto en el mundo analógico. ¿Habría llegado a la misma solución si hubiera podido usar etiquetas o búsqueda global en su Zettelkasten?

Repensando la función de los *hashtags* y analizando su conveniencia vemos que es posible adoptarlos como solución estándar para interconectar notas independientemente de la aplicación utilizada.

Anexo: gestión de notas desde línea de comandos

Para notas con etiquetas establecidas de esta manera no es necesario usar ninguna herramienta específica de gestión. Podría hacerse de forma simple usando **comandos UNIX** comunes desde un terminal.

Para los comandos que impliquen mostrar el contenido de las notas, una herramienta como Bat¹⁴ para el **resaltado de sintaxis** se convierte en una gran ayuda visual para mostrar Markdown, por ejemplo. Si estuviera instalada en el sistema, en los siguientes comandos se sustituiría `cat` por `bat`.

```
# Buscar las notas en las que aparezca la etiqueta #tag1:  
grep -wr '#tag1'
```

```
# Mostrar el contenido de todas las notas que contengan la etiqueta #tag1:  
grep -wrl '#tag1' | xargs cat
```

```
# Buscar todas las etiquetas que aparezcan en los archivos del  
# directorio actual y sus subdirectorios:  
grep -r '#[[:alnum:]]\+[[:space:]]'
```

```
# Tabla con el número de apariciones de cada etiqueta (tabla de frecuencias).  
grep -sorh '#[[:alnum:]]\+[[:space:]]' | sort | uniq -c | sort -rn
```

```
# Mostrar las notas asociadas a cada etiqueta  
grep -sor '#[[:alnum:]]\+[[:space:]]' | column -t -s':' -O 2,1 | sort | uniq
```

¹⁴<https://github.com/sharkdp/bat>

La salida de alguno de estos comandos puede servir como entrada para otras aplicaciones. Por ejemplo, si tenemos instalado WordCloud¹⁵ (`sudo apt install python3-wordcloud`) podemos generar una imagen que muestre una nube de etiquetas:

```
grep -sroh '#[[:alnum:]]\+[[:space:]]' \
| wordcloud_cli --imagefile tagcloud.png && eog tagcloud.png
```

Con estos comandos se puede crear un sencillo **lenguaje específico de dominio (DSL)** con los siguientes comandos:

```
tagtable      # Tabla de frecuencias de etiquetas.
tagcloud      # Nube de etiquetas.
tagnotes      # Notas asociadas a cada etiqueta.
tag '#tag1'   # Notas en las que aparezca la etiqueta #tag1
notes '#tag1' # Contenido de las notas que contengan la etiqueta #tag1
```

Para que el intérprete de comandos no considere el texto después de `#` como un comentario es necesario entrecomillar el nombre de la etiqueta. P.ej: `'#tag1'`

Lo codificamos añadiendo estas líneas al archivo `~/.bashrc`:

Zettelkasten DSL:

```
tagtable() { grep -sroh '#[[:alnum:]]\+[[:space:]]' \
| sort | uniq -c | sort -rn ; }

tagcloud() { grep -sroh '#[[:alnum:]]\+[[:space:]]' \
| wordcloud_cli --imagefile /tmp/tagcloud.png && eog /tmp/tagcloud.png ; }

tagnotes() { grep -sroh '#[[:alnum:]]\+[[:space:]]' \
| column -t -s':' -O 2,1 | sort | uniq ; }

tag() { grep -wr "$1" ; }

notes() { grep -wrl "$1" | xargs cat ; }
```

Una solución completa consistiría en agrupar estos y otros comandos y opciones en un script específico.

¹⁵https://github.com/amueller/word_cloud