

# El título de tu artículo

Juan Pérez  
Rancho Electrónico  
[juanp@rancho.org](mailto:juanp@rancho.org)

Leona Vicario  
Movimiento Independentista  
[leona.vicario@yahoo.mx](mailto:leona.vicario@yahoo.mx)

Fulanito  
Universidad Nacional Autónoma de México  
[fff@riseup.org](mailto:fff@riseup.org)

## ABSTRACT

Reemplaza este texto con un resumen de máximo 250 palabras. Debe ir indentado en el campo correspondiente del encabezado de metadatos en el documento markdown.

## 1 Introducción

Bienvenido a la plantilla para autores de la revista Heptagrama.



Este documento contiene una guía de uso de Markdown, a su vez escrita en Markdown, para elaborar propuestas de artículo. Para entender como se traduce Markdown a formato de publicación, necesitará comparar el archivo fuente `src/manual_de_autores.md` con `manual_de_editores.pdf`.

El uso de esta plantilla presupone un conocimiento básico del uso de la terminal de comandos.

Sugerimos hacer una copia de este documento, o mejor de todo el repositorio que lo contiene, como punto de partida para sus artículos.

Elaborar el artículo en Markdown permite enfocarse en el contenido del mismo, dejando a discreción del equipo editorial el formato y diseño del cuerpo del texto y bibliografía. Al separar el contenido de la forma, nuestro equipo editorial tiene la facilidad de adaptar y reconfigurar el formato de salida de los textos a conveniencia, ya sea para impresión (PDF) o web (HTML), a partir de un mismo archivo fuente.

## 2 Markdown

Con gusto podemos responder cualquier pregunta sobre Markdown y esta plantilla en relación a una propuesta de artículo vía [revistaheptagrama@gmail.com](mailto:revistaheptagrama@gmail.com).

Markdown es un lenguaje de marcado, en el mismo sentido que lo es HTML: los elementos de su sintaxis nos permiten *indicar el rol de cada objeto* dentro del texto final. Sin embargo, a diferencia de este último, Markdown está diseñado para que el texto plano siga siendo legible. Corresponde al paradigma WYSIWIM (*What You See Is What I Mean*), en el que se indica de manera declarativa el formato de cada objeto del texto, separando de manera efectiva el contenido de su formato de impresión (tipografía, distribución, etc.).

Aprender Markdown tiene un valor independiente de la aplicación a la revista; se trata de una tecnología ubicua en una gran variedad de contextos (plataformas de código, diseño web, aplicaciones de notas inteligentes) que otorga facilidades accesibles para hacer publicaciones profesionales fáciles de procesar y migrar.

### 2.1 Pandoc Markdown

Markdown tiene diversas variantes. Aquí utilizamos el Pandoc Markdown que incluye una gran colección de extensiones que permiten especificar diversas estructuras de formato. Como referencia general puede consultar el [manual de Pandoc Markdown](#); en este documento revisaremos el marcado de algunos tipos de formato básicos.

### 2.2 Markdown a PDF

Para producir un PDF a partir del documento fuente en Markdown podemos ejecutar en una terminal el siguiente comando:

```
pandoc --template=pandoc/heptagrama.latex \
--citeproc --number-sections manual_de_editores.md \
-o manual_de_editores.pdf
```

Este comando presupone que la sesión de terminal se encuentra en el repositorio, por lo que usamos su ubicación relativa. En general la opción `template` debe apuntar a la ubicación del archivo `heptagrama.latex` y `manual_de_editores.md` corresponde a la ubicación del archivo fuente.

Para un documento de mayor calidad añadimos la opción `--pdf-engine=xelatex` al comando anterior.

### 2.2.1 Dependencias

Para que el comando anterior funcione es necesario instalar:

- Pandoc. [Instrucciones oficiales de instalación](#).
  - MacOS: Se recomienda utilizar el administrador de paquetes [Homebrew](#). Correr `brew update` en la terminal y en seguida `brew install pandoc`.
  - Windows: Se recomienda utilizar el administrador de paquetes [Chocolatey](#). Correr en terminal `choco install pandoc`.
  - Linux: usar el administrador de paquetes de la distribución.
- LaTeX (se puede instalar de manera similar utilizando los administradores de paquetes antes mencionados). **No se manipulará ningún archivo LaTeX, Pandoc lo utiliza internamente para procesar los documentos.** Se recomienda las siguientes distribuciones: [MacTeX](#) para MacOS, [MiKTeX](#) o [Tex Live](#) para Windows y [Tex Live](#) para linux.
  - Asegurarse de tener instalado XeTeX en caso de tener una instalación parcial de la distribución de LaTeX (particularmente en el caso de Tex Live en Linux).
- Fuentes:
  - [Linux Libertine](#)
  - [Inconsolata](#)

Si además se instala `make`,<sup>1</sup> entonces se puede ejecutar `make pdf` o `make html` en lugar de los comandos de Pandoc (definidos en el archivo `Makefile`).

## 3 Estructura

Los títulos de sección y subsección, así como cualquier jerarquía de títulos, se marca utilizando numerales/sostenidos (`#`, `##`, `###`, ...) al principio de la línea correspondiente. Estos títulos deben estar en su propio párrafo.

Como se observa aquí, los saltos de línea se interpretan como espacios.

---

<sup>1</sup><https://www.gnu.org/software/make/>

Los párrafos y los títulos deben estar aislados por líneas vacías; múltiples líneas vacías se interpretan como una sola como se observa al final de esta sección (más líneas vacías no aumentan el espacio entre párrafos o títulos).

Para crear notas al pie de página se ubica su contenido en otro párrafo etiquetado con una palabra de referencia o identificador.<sup>2</sup> También se puede incluir la nota en la misma línea sin necesidad de identificador.<sup>3</sup>

## 4 Énfasis

Así mismo podemos hacer de un texto **negrita**, *cursiva*, o ***ambas*** para dar énfasis a oraciones o palabras. *No debe haber espacio entre las marcas y el principio y final de la palabra u oración.*

## 5 Citas

Se puede incluir [bibliografía añadiendo elementos en formato biblatex](#) al archivo `referencias.bib`. Estas pueden ser generadas de manera automática por software de gestión bibliográfica como [Zotero](#); también es posible hacerlo a mano o utilizando algunas herramientas en línea. Con los siguientes pasos se añade una referencia para ser citada en el texto:<sup>4</sup>

1. Añadir la entrada correspondiente en formato *BibLaTeX*, a `referencias.bib`. Ejemplo:

```
@book{Blackwell2022Live,
  title = {Live Coding: A User's Manual},
  shorttitle = {...},
  author = {...},
  date = {...},
  series = {...},
  publisher = {...},
  location = {...},
  url = {...},
  abstract = {...},
  isbn = {...},
  pagetotal = {...},
  keywords = {...}
}
```

2. Utilizar la llave de referencia indicada al principio de la entrada. En el ejemplo es `Blackwell2022Live`.
3. Llamarla dentro del texto dependiendo del caso

---

<sup>2</sup>Esta es una nota al pie de página. La primera línea de los párrafos subsecuentes están indentados para indicar que pertenecen a la misma nota.

La llave identificador utilizada para esta nota es una palabra arbitraria que permite identificar el pie de página; equivalentemente podemos usar un número. Al renderizar cada pie estará indicado por un número correspondiente a su ubicación en la secuencia de notas.

<sup>3</sup>En este caso, la nota al pie de página está indicado en la misma ubicación que la refiere. De esta forma no requiere un identificador, pero no puede tener múltiples párrafos.

<sup>4</sup>Sintaxis de citas en el manual de Pandoc.

1. @Blackwell12022Live sale como Blackwell et al. (2022).
2. [@Blackwell12022Live] sale como (Blackwell et al. 2022).
3. [@Blackwell12022Live; @Magnusson2018Performing] sale como (Blackwell et al. 2022; Magnusson and McLean 2018).

Por ejemplo, Blackwell et al. describen la recursión temporal como una estrategia para codificar el cambio de una estructura musical en el tiempo:

The 2011 Extempore system and its 2005 precursor Impromptu, both by Andrew Sorensen, are similar in that music is declaratively represented, using step-by-step structures associated with imperative programming. A core feature of Extempore is temporal recursion, in which a function calls itself with a particular time delay. This uses what is known as tail recursion, in which the final step in a function is the recursive call to itself. Temporal recursion is Extempore's solution to a problem that every live coding environment has to face: How can a program be interpreted while it is itself undergoing change? A temporally recursive function always runs to the end, but if it is replaced in the meantime, it then recursively calls this new version of itself. Temporal recursion therefore manages both change and time. (2022, chap. 6).

Esta y otras aproximaciones teóricas a la música están mediadas por un contexto que el investigador en música valida o cuestiona:

La existencia de un marco teórico no es una exclusiva de los textos científicos. El marco teórico existe siempre, seamos o no conscientes de ello. Hablar de «período clásico» en relación con las obras de Haydn y Mozart o argumentar en torno a la fascinación de Bartók por la «música popular» por ejemplo, supone hacer nuestra una terminología cuyo origen posiblemente desconocemos y que puede ser objeto de fuertes discrepancias. Incluso el más humilde de los textos de divulgación contribuye a consolidar antiguas certezas, si las asume como propias. Por el contrario, si al escribir tomamos una posición a contracorriente, aunque avalada por la opinión de otros investigadores de prestigio, sin duda estaremos contribuyendo a difundir nuevos planteamientos que, al menos hoy, nos parecen más adecuados. (Chiantore, Domínguez Moreno, and Martínez García 2016, 24)

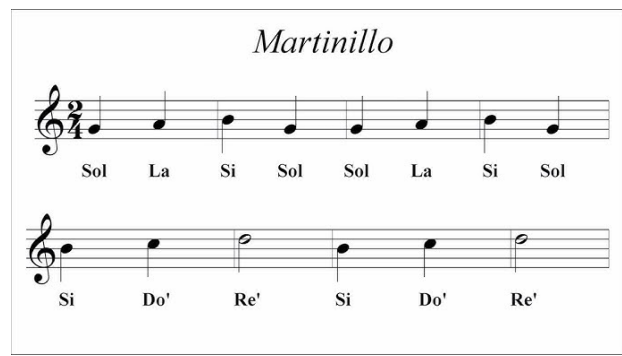


Figure 1: Partitura de “Martinillo”

En `references.bib` podemos ver ejemplos de artículo de revista (Toussaint 2005), de artículo de memorias de coloquio (Melkonian et al. 2019), libros (Blackwell et al. 2022; Chiantore, Domínguez Moreno, and Martínez García 2016) y capítulo de libro (Magnusson and McLean 2018). Existen más categorías cuya elección dependerá del material referenciado. Las referencias correspondientes a las citas llamadas en el texto serán desplegadas automáticamente al final del artículo, después del título vacío **Referencias**.

## 6 Imágenes y enlaces

Podemos incluir enlaces internos y externos en el texto. Por ejemplo, enlazar a la **sección sobre Markdown** o una página de internet, como este **manual de Markdown en español**. Para incluir el enlace explícitamente, utilizamos la misma sintaxis, repitiendo la dirección: <https://sites.google.com/view/heptagrama-unam/>.

La sintaxis para incluir imágenes es similar: basta con agregar `!` al inicio del enlace. Por ejemplo, si ubicamos las imágenes en `/src/img/`, y queremos incluir la imagen de nombre `imagen.jpg` la incluimos así:

```
![Todas las imágenes deben incluir una
descripción detallada, entendible sin referencia
al resto del artículo](img/imagen.jpg).
```

Si la llamada a la imagen es lo único en su párrafo, entonces se incluirá como figura y se acomodará en el texto de manera automática. En este caso el texto de la liga se muestra como la descripción de la figura. También podemos referir a una figura utilizando su etiqueta, por ejemplo la **Figura 1**.

De otra forma dicho texto queda como texto alternativo de la imagen (que, en el caso de HTML, significa que sólo se muestra cuando la página web no logra cargarla). Por ejemplo, una imagen en línea con atributos (modificando su tamaño): 🙌.

## 7 Bloques de Código

Podemos hacer que el texto adquiriera fuente monoespacio (o de **código**) utilizando el símbolo ‘ como delimitador.

Logramos lo mismo para un párrafo indentándolo cuatro espacios o un TAB

Una opción más flexible es insertar tres o más ‘ para producir un bloque de código en el que podemos indicar el lenguaje de programación para el marcado automático de su sintaxis:

```
factorial :: Int -> Int
factorial 0 = 1
factorial n = n * factorial (n - 1)
```

Equivalentemente podemos delimitarlo utilizando tildes ~, e incluir atributos para el formato del mismo (aunque no todos estos atributos aplican para algunos los formatos de salida). Se necesitan mínimo 3 tildes en la línea de apertura y más o igual en la línea de cierre:

```
100 quickSort :: Ord a => [a] -> [a]
101 quickSort []      = []
102 quickSort (x:xs) = quickSort (filter (< x) xs)
103                   ++ [x] ++
104                   quickSort (filter (>= x) xs)
```

Para más referencia consultar la sección [Verbatim \(code\) blocks](#) del manual de Pandoc.

## 8 Música

Para incluir música en pentagrama podemos generarla desde una aplicación externa e insertarla como imagen en la forma que vimos [antes](#). Como ejemplo tenemos este fragmento de la Feldpartita de Haydn en si $\flat$ :



Como alternativa al uso de imágenes para incluir música en pentagrama, se puede utilizar la [notación abc](#) en un bloque de código. Tras bambalinas este código es utilizado para generar la imagen en cuestión. Para ello es necesario tener instalado [absm2ps](#).

Speed the Plough

Trad.



## 9 Matemáticas


Se pueden incluir fórmulas matemáticas en línea,  $\sqrt{3x-1} + (1+x)^2$ , o en su propio párrafo

$$\int_a^b x^2 dx$$

$$\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2 \right) \left( \sum_{k=1}^n b_k^2 \right)$$

utilizando el delimitador \$ y la notación de LaTeX.

## 10 Conclusiones

Esperamos recibir sus propuestas terminadas en [revistaheptagrama@gmail.com](mailto:revistaheptagrama@gmail.com). Notificaremos de las diversas etapas de revisión y tiempos de respuesta. En Heptagrama la revisión por pares es doble ciego siguiendo los lineamientos de publicación [Open Access](#) . Para más información consultar nuestras [políticas editoriales](#).

## Referencias

10 Blackwell, Alan F., Emma Cocker, Geoff Cox, Alex McLean, and Thor Magnusson. 2022. *Live Coding: A User's Manual*. Software Studies. Cambridge, Massachusetts: The MIT Press. <https://livecodingbook.toplap.org/>.

Chiantore, Luca, Áurea Domínguez Moreno, and Sílvia Martínez García. 2016. *Escribir Sobre Música*. Primera edición. Barcelona: Musikeon Books.

Magnusson, Thor, and Alex McLean. 2018. “Performing with Patterns of Time.” In *Oxford Handbook of Algorithmic Music*. Oxford University Press. <https://doi.org/10.5281/zenodo.1193251>.

Melkonian, Orestis, Iris Yuping Ren, Wouter Swierstra, and Anja Volk. 2019. “What Constitutes a Musical Pattern?” In *Proceedings of the 7th ACM SIGPLAN International Workshop on Functional Art, Music, Modeling, and Design*, 95–105. Berlin Germany: ACM. <https://doi.org/10.1145/3331543.3342587>.

Toussaint, Godfried. 2005. “The Euclidean Algorithm Generates Traditional Musical Rhythms.” *Proceedings of BRIDGES: Mathematical Connections in Art, Music and Science*, 47–56.