

Universidad Tecnológica Nacional

Facultad Regional Buenos Aires

Escuela de Posgrado

MAESTRÍA EN INGENIERÍA EN SISTEMAS DE
INFORMACIÓN

Dir: Dra. Ma. Florencia Pollo Cattaneo

Seminario

HERRAMIENTAS PARA EL DESARROLLO DE TESIS 2021

Prof: Florencia Pollo Cattaneo

Trabajo Práctico Final

PRODUCCIÓN ACADÉMICA CON PANDOC

Lisandro Fernández

Abstract

Pandoc como entorno textutal de producción de documentos académicos. Evitar el uso de interfaces captivas beneficia a todos los usuarios, deben poder encontrar lo que necesitan, comprender lo que encuentran y usarlo para realizar tareas.’

Octubre 2022

Buenos Aires, Argentina

Contendios

1	Producción gráfica de documentos Pandoc	1
1.1	Pandoc	1
1.2	Texto Plano	2
2	Medodotología	3
2.1	Integración	3
2.2	Gráficos y diagramas	4
2.3	BibLATEX - Bibliografías sofisticadas en LATEX	7
2.4	MathJax	8
3	Resultados	9
3.1	Cheat Shet and examples	9
4	Conclusión	9
4.1	Futuras lineas de trabajo	10
4.2	Debilidades	10
4.3	Indicaciones	11
	Referencias	11

1 Producción gráfica de documentos Pandoc

Este proyecto describe el proceso de confección de escritos académicos, de complejidad considerable, sin la necesidad de interfaces gráficas [1]. Cuenta con capacidades necesarias para la producción de documentación técnica y científica, todo el proceso es controlado mediante linea de comandos evitando así depender de interfaces captivas [2, pp. 88–97]

El objetivo de este trabaja es un entorno el cual *Pandoc* “*a universal document converter*” es la pieza central, interprete del sistema de composición tipográfica y preparación de documentos de alta calidad LaTeX, estándar de facto para la comunicación y publicación de documentos académicos [3, 4].

Mediante integraciones sencillas se consigue una infraestructura robusta con funciones diseñadas para gestionar exposición de extensas referencias, múltiples citas y bibliografía a diferentes fuentes; notación matemática, generación gráficos y diagramas, entre otras capacidades.

1.1 Pandoc

Pandoc es una biblioteca *Haskell* [5] para convertir de un formato de marcado ligero a otro, y una herramienta de línea de comandos que accede a las funciones en esta biblioteca para convertir entre numerosos formatos y de procesamiento de textos.

La biblioteca incluye módulos separados para cada formato de entrada y salida, por lo que para añadir un nuevo formato de entrada o salida sólo hay que añadir un nuevo módulo.

1.2 Texto Plano

Evitar el uso de formatos codificados o de poca legibilidad e interfaces captivas beneficia a todos los usuarios, que deben poder encontrar lo que necesitan, comprender lo que encuentran y usarlo para realizar tareas [6]. Habilita el separar contenido, referencias y estilo.

En un contexto de organizaciones con actividades relacionadas a la publicación donde la complejidad no solo de documentos sino que también lo es la tarea ya involucra a múltiples agentes como autores, correctores y editores,

el control estético y da unidad en estilo en la composición gráfica resultante de diversos productos.

De los beneficios que trabajar con texto plano habilita se resalta la accesibilidad y la posibilidad de producir documentos gráficos de la misma manera que se produce software

incluirllo en una cadena de desarrollo de propia de las software factories

y viceversa es un learn by example para escritores/editores pueden adquirir capacidad de flujo de trabajo

se señala como futuras líneas de trabajo es fácilmente incluíble en operaciones remotas automáticas.

1.2.1 Markdown

Markdown es una sintaxis plain text formatting syntax de formato de texto,

El formato de texto es el marcado que se aplica a un texto simple para añadir datos de estilo más allá de la semántica de los elementos: colores, estilos (negrita, itálica), tamaño, y características especiales (como hipervínculos). Al texto resultante se le conoce como texto formateado, texto con estilos, o texto enriquecido

<https://wires.onlinelibrary.wiley.com/doi/10.1002/wics.1348>

This article is categorized under: Software for Computational Statistics > Software/Statistical Software

El con énfasis en la legibilidad

de autoría de texto Lo que distingue a Markdown de muchas otras sintaxis de marcado ligero, es su legibilidad.

Como escribe Gruber: El objetivo primordial del diseño de la sintaxis de formato de Markdown es hacerla lo más legible posible. La idea es que un doc-

umento con formato Markdown sea publicable tal cual, como texto plano, sin que parezca que ha sido marcado con etiquetas o instrucciones de formato. (<http://daringfireball.net/projects/markdown/>)

<https://github.github.com/gfm/>

La versión mejorada de Markdown de Pandoc incluye sintaxis para tablas, listas de definiciones, bloques de metadatos, notas a pie de página, citas, matemáticas y mucho más.

Véase más abajo en Markdown de Pandoc. <https://pandoc.org/MANUAL.html#pandocs-markdown>

Pandoc comprende una serie de extensiones útiles de la sintaxis de markdown, como los metadatos del documento (título, autor, fecha); las notas al pie; las tablas; las listas de definiciones; los superíndices y subíndices; la tachadura; las listas ordenadas mejoradas (el número de inicio y el estilo de numeración son significativos); las listas de ejemplos en ejecución; los bloques de código de software delimitados con resaltado de sintaxis; las comillas inteligentes, los guiones y las elipses; el markdown dentro de bloques HTML; y el LaTeX en línea.

Si se desea una compatibilidad estricta con markdown, todas estas extensiones pueden desactivarse.

2 Metodología

El desarrollo de este trabajo fue realizado nota al pie: Como conseguir una instalacion funcional de Esta fuera de los alcances de este articulo.

un potente sistema para escribir filtros. El diseño modular: consta de un conjunto de lectores, que analizan el texto en un formato determinado y producen una representación nativa del documento (Abstract Sintactic Three - AST), y un conjunto de escritores, que convierten esta representación nativa en un formato de destino.

De los múltiples maneras de personalizar *Pandoc* para que se adapte a los requisitos de cada proyecto, se destaca el uso de un sistema de plantillas y un potente sistema de citas y bibliografías automáticas y la generación de gráficos mediante código.

2.1 Integración

bibliografia

Por ejemplo, esto significa que puede escribir una referencia como @moolenaar2000 o también [knuth1986texbook p.3-9] y *Pandoc* a convertirá en una cita con el formato predefinido, utilizando cualquiera de los cientos de Lenguajes de Estilo de Cita (Citation Style Language - CSL)

incluyendo estilos de nota al pie, numéricos y autoría, fuente y fechas), y añadirá una bibliografía con el formato adecuado al final del documento.

Las matemáticas de LaTeX (e incluso las macros) pueden utilizarse en los documentos de *Markdown*. Las matemáticas de LaTeX se convierten (según lo requiera el formato de salida) en unicode, objetos de ecuación nativos de Word, MathML o roff eqn.

sistema de diagramacion y generacion graficos

permite crear diagramas y visualizaciones utilizando texto y código. Se trata de una herramienta de diagramación y graficación que renderiza definiciones de texto inspiradas en Markdown para crear y

modificar diagramas de forma dinámica.

2.2 Gráficos y diagramas

Diagramación y la documentación cuestan un tiempo precioso a los desarrolladores y quedan obsoletas rápidamente.

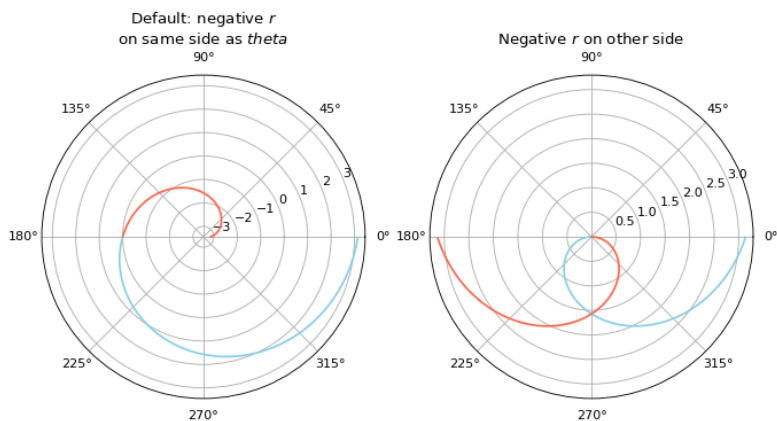
Pero no tener diagramas o documentación arruina la productividad y perjudica el aprendizaje de la organización.

2.2.1 pandoc-plot

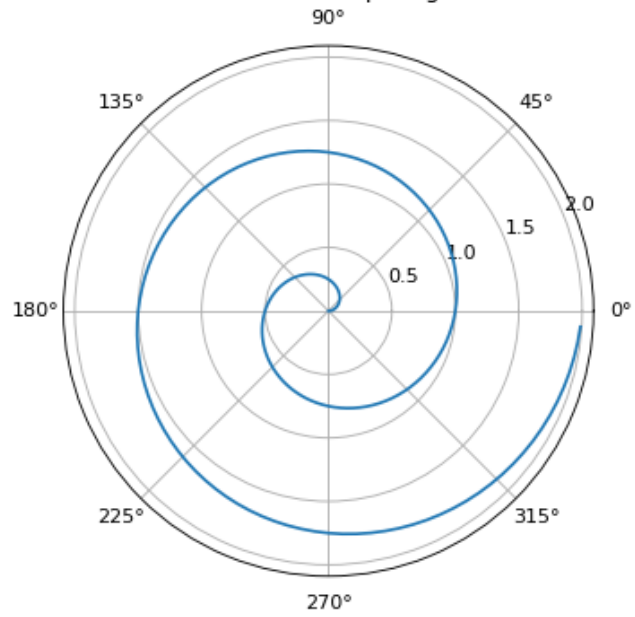
<https://laurenttrdc.github.io/pandoc-plot/>

pandoc-plot es un filtro de *Pandoc* para generar figuras a partir de bloques de código en documentos.

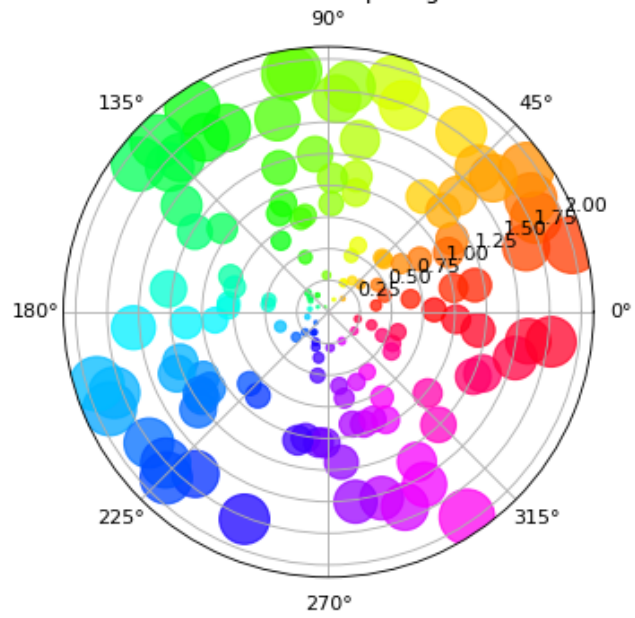
Esta página está generada a partir de *Markdown* utilizando *pandoc-plot*, para que puedas hacerte una idea de lo que es posible.

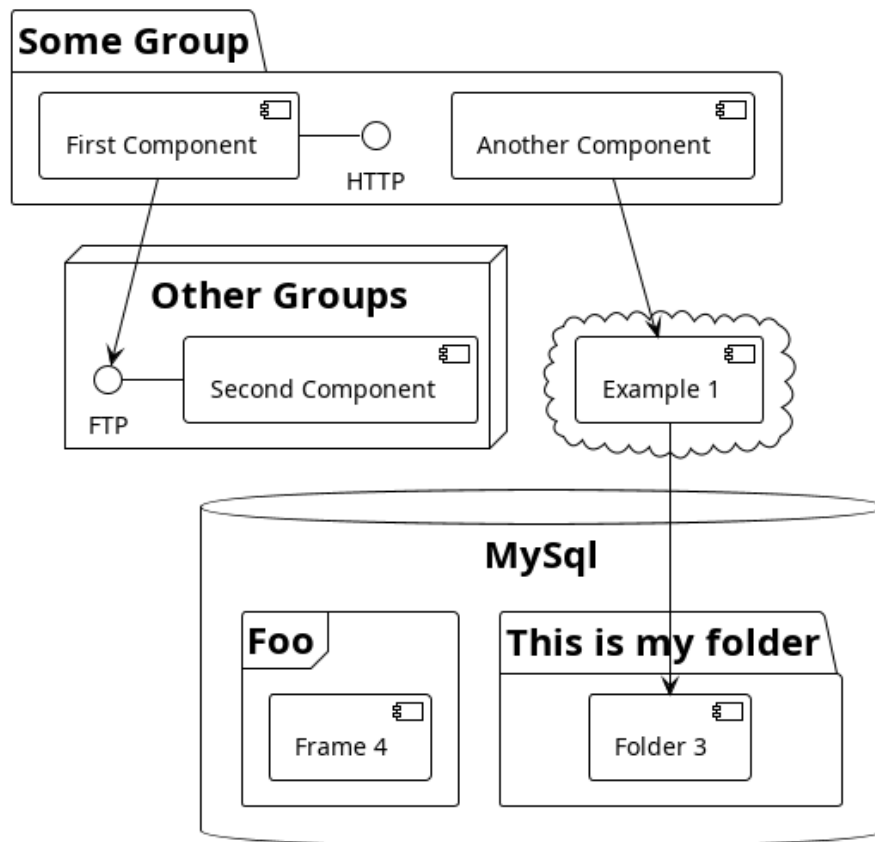


This is an example figure

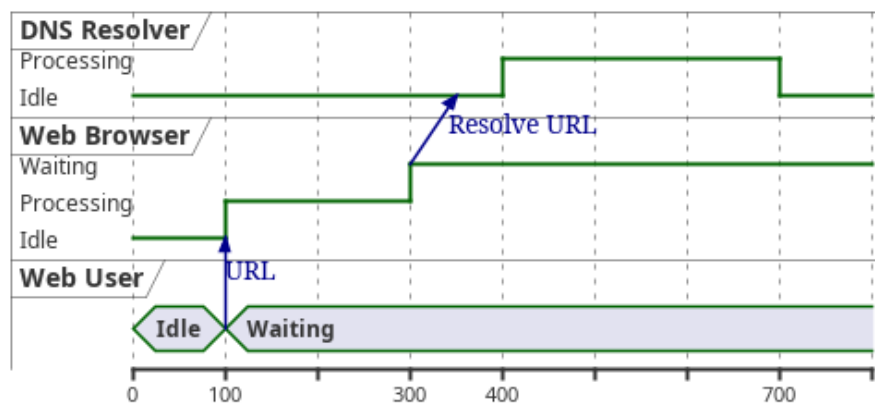


This is an example figure





<https://the-lum.github.io/puml-themes-gallery/>



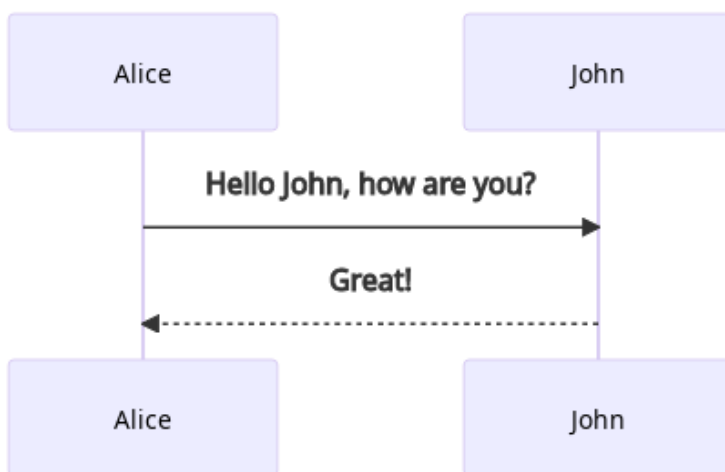
2.2.2 MermaidJS

Mermaid es una herramienta de diagramación y gráficos basada en JavaScript que utiliza definiciones de texto inspiradas en Markdown y un renderizador para crear y modificar diagramas complejos. El objetivo principal de Mermaid es ayudar a que la documentación se ponga al día con el desarrollo.

Mermaid aborda este problema permite crear diagramas fácilmente modificables, del código de piezas de software.

Mermaid permite incluso a los no programadores crear fácilmente diagramas detallados y a través del Mermaid Live Editor.

```
~~~mermaid
sequenceDiagram
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
~~~
```



2.3 BibLATEX - Bibliografías sofisticadas en LATEX

Los datos bibliográficos pueden estar en formato BibTeX, BibLaTeX, CSL JSON o CSL YAML. Las citas funcionan en todos los formatos de salida.

BibLaTeX es una reimplementación completa de las facilidades bibliográficas proporcionadas por LaTeX. El formato de la bibliografía está totalmente controlado por las macros de LaTeX, y un conocimiento práctico de LaTeX debería ser suficiente para diseñar nuevos estilos de bibliografía y citación.

Con biber BibLATEX tiene muchas características que rivalizan o superan a otros sistemas bibliográficos.

2.3.1 CLS

<https://www.mendeley.com/guides/csl-editor/>

El Lenguaje de Estilo de Citación (CSL) es un formato basado en XML para describir el formato de citas, notas y bibliografías, ofreciendo:

<https://docs.citationstyles.org/en/stable/primer.html#what-is-csl>

Para obtener más documentación, el esquema CSL, los estilos y las localizaciones, visite la página web del proyecto CSL, citationstyles.org.

Si alguna vez has escrito un trabajo de investigación, habrás trabajado de investigación,

probablemente has incluido referencias a otros trabajos. Las referencias son importantes en la comunicación académica, ya que proporcionan la atribución, enlazan referentes.

Sin embargo, formatear manualmente las referencias puede llevar mucho tiempo, especialmente cuando se trata de múltiples publicaciones con diferentes estilos de citación.

El software de gestión de referencias puede ayudar. no sólo ayudan a gestionar bibliotecas de investigación, sino que también pueden generar automáticamente citas y bibliografías. Pero para formatear las referencias en el estilo deseado, estos programas necesitan descripciones de cada estilo de citación en un lenguaje que el ordenador pueda entender. Como habrás adivinado, el Lenguaje de Estilos de Citación (CSL) es ese lenguaje

<https://docs.citationstyles.org/en/stable/specification.html>

2.4 MathJax

Se proporcionan varios métodos diferentes para representar las matemáticas en HTML, incluyendo MathJax y la traducción a MathML.

Cuando $a \neq 0$, hay dos soluciones a $(ax^2 + bx + c = 0)$ las cuales son

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

Transformación de contenidos: Logotipo estilizado de EpubMathJax proporciona herramientas para transformar sus contenidos de fuentes impresas tradicionales en contenidos web y ePubs modernos y accesibles.

Tipografía de alta calidad: MathJax utiliza CSS con fuentes web o SVG, en lugar de imágenes de mapa de bits o Flash, por lo que las ecuaciones se escalan con el texto circundante en todos los niveles de zoom.

Modular la entrada y la salida: MathJax es altamente modular en la entrada y la salida. Utiliza MathML, TeX, y ASCIImath como entrada y produce HTML+CSS, SVG, o MathML como salida.

Accesible y reutilizable: MathJax funciona con lectores de pantalla y proporciona zoom de expresión y exploración interactiva. También puede copiar ecuaciones en Office, LaTeX, wikis y otro software.

3 Resultados

El producto de este proyecto la integración de diferentes piezas de software y andamiaje necesario para reproducir este proyecto esto incluye: Configuraciones, estructura ejemplo filtros, cls, plantilla LaTeX, resaltado de código, configuraciones para operaciones remotas

Un ejemplo de como usar esta herramienta es el mismo comando que sirve para conseguir este documento desde su fuente en *Markdown* [**markdown?**].

```
$ pandoc README.md \
-F pandoc-plot --metadata-file=metadata.yaml --mathjax \
-F mermaid-filter -F pandoc-crossref --citeproc \
--highlight-style pygments.theme \
--template=plantilla --pdf-engine-opt=-shell-escape \
-s --toc --toc-depth=2 --number-sections --columns=80 \
-o README.pdf
```

Ejecutar el comando precedente en una sistema con una instancia de *Pandoc* funcionando produce las este documento y declara: plantillas: General LaTeX, lenguaje de estilo de citas, resaltado de código. Módulos extra como filtros: Notación matemática, gestión de Referencias, citas y generación de gráficos. Entre otras configuraciones generales involucradas en el proceso, la profundidad la tabla de contenidos. archivo de entrada y de salida.

3.1 Cheat Shet and examples

apendice.pdf

El documento y su contraparte

Para recrear el proceso que se utilizo para confeccionar este documento

4 Conclusión

Se consigue un cadena de producción

Características generales: un formato libre y abierto, componentes aislados, compactos y robustos; amplia compatibilidad con requisitos de estilo, miles de estilos disponibles predifindos por la comunidad, etiquetas de bibliografía altamente

personalizables y fuentes de datos remotas para publicaciones frecuentes con información dinámico.

Citas y citas: localización automática de estilos, infraestructura para la distribución y actualización de estilos, compleja modificación de datos sobre la producción y basada en macros sin cambiar las fuentes de datos.

4.1 Futuras líneas de trabajo

Se señalan algunas áreas de desarrollo

4.1.1 Operaciones remotas automáticas

<https://pandoc.org/installing.html#github-actions>

disponible el repositorio con



Integración continua <https://about.gitlab.com/features/continuous-integration/>

4.1.2 Revisión sistemática de literatura

Para revisiones de literatura que comprometan múltiples bibliografías y listas de información bibliográfica en el mismo documento con diferente ordenación

- Ordenación altamente personalizable mediante el Algoritmo de Cotejo Unicode (Unicode collation algorithm - UCA) (<https://www.unicode.org/reports/tr10/>)
- Adaptación de CLDR (<https://cldr.unicode.org/>)
- Bibliografías jeraquizadas por capítulo, sección, etc.
- Soporte de poliglosia para el cambio automático de idioma de las entradas y citas bibliográficas
- Modelo de datos altamente personalizable para que los usuarios puedan definir sus propios tipos de datos bibliográficos
- Validación de los datos bibliográficos con respecto a un modelo de datos

4.2 Debilidades

Dado que la representación intermedia de un documento por parte de Pandoc es menos expresiva que muchos de los formatos entre los que convierte, no hay que esperar conversiones exactas entre todos los formatos. Mientras que las conversiones de Markdown de Pandoc a todos los formatos aspiran a ser perfectas, las conversiones de formatos más expresivos que Markdown de Pandoc pueden tener diferencias.

Pandoc intenta conservar los elementos estructurales de un documento, pero no los detalles de formato, como el tamaño de los márgenes.

Y algunos elementos del documento, como por ejemplo tablas complejas, pueden no encajar en el modelo de documento simple de Pandoc.

Si bien los módulos empleados maduraron y se distribuyen dentro con empaquetados la distribución principal de Pandoc cambiar configuraciones agregar funcionalidades y rutinas puede variar y depender de versiones de las herramientas en las diferentes distribuciones y sistemas operativos puede inferir en la integración y operaciones.

Pero en una implementación organizacional esto puede ser solucionado ejecutando en servidor remoto donde como un servicio de preparación de documentos gráficos, tanto preparando para impresión o para ser consumidos mediante exploradores de internet solamente se carguen como insumos los ficheros de contenido y las configuraciones y plantillas estén

4.3 Indicaciones

Es intencional que este proyecto que sirva como plantilla para proyectos similares, patrones de diseño y guía de buenas prácticas para la producción de documentos gráficos de alta complejidad.

y siendo que este proyecto está enfocado a la producción de literatura académica, esta misma cadena puede ser considerada en el desarrollo de cualquier sistema de gestión documental, registros médicos, documentos legales, certificados, etc.

Donde productos gráficos imprimibles se generan mediante rutinas directamente de bases de datos, una capa codificada extra que opaca la relación con el interprete el y el contenido, se recomienda un proceso similar al descrito de respaldo de la información en contenedores de formato simple y legible, sin codificar u opacar habilitando ser manipulados con herramientas obvias, con acceso directo mediante sistemas rudimentales.

Referencias

- [1] W. Caleb McDaniel, “Why (and how) i wrote my academic book in plain text – w. Caleb McDaniel,” *W. Caleb McDaniel*. Available: <http://wcaleb.org/blog/my-academic-book-in-plain-text>
- [2] M. Gancarz, *Linux and the unix philosophy*. Elsevier Science, 2003. Available: <https://books.google.com.ar/books?id=qqstCSlk5MIC>
- [3] John MacFarlane, “Pandoc - a universal document converter,” *Pandoc - a universal document converter*. 2022. Accessed: Sep. 14, 2022. [Online]. Available: <https://pandoc.org/>
- [4] D. E. Knuth, D. Knuth, and D. Bibby, *The TeXbook*. Addison-Wesley, 1986. Available: www-cs-faculty.stanford.edu/~knuth/abcde.html
- [5] S. Marlow *et al.*, “Haskell 2010 language report,” *Available online http://www.haskell.org/(May 2011)*, 2010.
- [6] D. A. S. U. Harvard, “Use plain language,” *Digital Accessibility*. Digital Accessibility Services. Available: accessibility.huit.harvard.edu/use-plain-language

- [7] B. Moolenaar, “Seven habits of effective text editing,” 2000. mooleenaar.net/habits.html
- [8] J. Webster and R. T. Watson, “Analyzing the past to prepare for the future: Writing a literature review,” *MIS Quarterly*, vol. 26, no. 2, 2002, Available: <http://www.misq.org/misreview/announce.html>
- [9] B. Kitchenham, B. Kitchenham, and S. Charters, “Guidelines for performing Systematic Literature Reviews in Software Engineering,” -, 2007, doi: [10.1.1.117.471](https://doi.org/10.1.1.117.471).
- [10] R. Wieringa, N. Maiden, N. Mead, and C. Rolland, “Requirements engineering paper classification and evaluation criteria: A proposal and a discussion,” *Requirements Engineering*, vol. 11, no. 1, pp. 102–107, Mar. 2006, doi: [10.1007/s00766-005-0021-6](https://doi.org/10.1007/s00766-005-0021-6).
- [11] T. Aljowder, M. Ali, and S. Kurnia, “Systematic literature review of the smart city maturity model,” *2019 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2019*, Sep. 2019, doi: [10.1109/3ICT.2019.8910321](https://doi.org/10.1109/3ICT.2019.8910321).
- [12] N. S. N. Wahab, T. W. Seow, I. S. M. Radzuan, and S. Mohamed, “A Systematic Literature Review on The Dimensions of Smart Cities,” *IOP Conference Series: Earth and Environmental Science*, vol. 498, no. 1, p. 012087, May 2020, doi: [10.1088/1755-1315/498/1/012087](https://doi.org/10.1088/1755-1315/498/1/012087).
- [13] J. Stübinger and L. Schneider, “Understanding Smart City—A Data-Driven Literature Review,” *Sustainability 2020, Vol. 12, Page 8460*, vol. 12, no. 20, p. 8460, Oct. 2020, doi: [10.3390/SU12208460](https://doi.org/10.3390/SU12208460).
- [14] J. F. Wolfswinkel, E. Furtmueller, and C. P. M. Wilderom, “Using grounded theory as a method for rigorously reviewing literature,” <https://doi.org/10.1057/ejis.2011.51>, vol. 22, no. 1, pp. 45–55, 2017, doi: [10.1057/EJIS.2011.51](https://doi.org/10.1057/EJIS.2011.51).
- [15] C. Okoli, “A Guide to Conducting a Standalone Systematic Literature Review,” *Communications of the Association for Information Systems*, vol. 37, no. 1, p. 43, Nov. 2015, doi: [10.17705/1CAIS.03743](https://doi.org/10.17705/1CAIS.03743).
- [16] B. A. Kitchenham, D. Budgen, and O. P. Brereton, “Using mapping studies as the basis for further research – A participant-observer case study,” *Information and Software Technology*, vol. 53, no. 6, pp. 638–651, Jun. 2011, doi: [10.1016/j.infsof.2010.12.011](https://doi.org/10.1016/j.infsof.2010.12.011).
- [17] D. Budgen, M. Turner, P. Brereton, and B. Kitchenham, “Using Mapping Studies in Software Engineering,” 2008, Available: www.ebse.org.uk.
- [18] J. Heaton and A. K. Parlikad, “A conceptual framework for the alignment of infrastructure assets to citizen requirements within a Smart Cities framework,” *Cities*, vol. 90, pp. 32–41, Jul. 2019, doi: [10.1016/J.CITIES.2019.01.041](https://doi.org/10.1016/J.CITIES.2019.01.041).

- [19] K. A. Achmad, L. E. Nugroho, A. Djunaedi, and Widyawan, "Smart City for Development: Towards a Conceptual Framework," *Proceedings - 2018 4th International Conference on Science and Technology, ICST 2018*, Nov. 2018, doi: [10.1109/ICSTC.2018.8528677](https://doi.org/10.1109/ICSTC.2018.8528677).
- [20] N. Guarino, G. Guizzardi, and J. Mylopoulos, "On the Philosophical Foundations of Conceptual Models," *Frontiers in Artificial Intelligence and Applications*, vol. 321, pp. 1–15, Dec. 2020, doi: [10.3233/FAIA200002](https://doi.org/10.3233/FAIA200002).
- [21] H. C. Mayr and B. Thalheim, "The triptych of conceptual modeling," *Software and Systems Modeling*, vol. 20, no. 1, pp. 7–24, Nov. 2020, doi: [10.1007/S10270-020-00836-Z](https://doi.org/10.1007/S10270-020-00836-Z).
- [22] L. M. L. Delcambre, S. W. Liddle, O. Pastor, and V. C. Storey, "Characterizing conceptual modeling research," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11877 LNCS, pp. 40–57, 2019, doi: [10.1007/978-3-030-33246-4_3](https://doi.org/10.1007/978-3-030-33246-4_3).
- [23] J. H. Coombs, A. H. Renear, and S. J. DeRose, "Markup Systems and the Future of Scholarly Text Processing," *Commun. ACM*, vol. 30, no. 11, pp. 933–947, Nov. 1987, doi: [10.1145/32206.32209](https://doi.org/10.1145/32206.32209).
- [24] J. Wild, "A Review of the Humdrum Toolkit: UNIX Tools for Musical Research, created by David Huron," *Music Theory Online*, vol. 2, no. 7, 1996.
- [25] E. Selfridge-Field, *Beyond MIDI: The Handbok of Musical Codes*. Estados Unidos: The MIT Press, 1997.
- [26] B. W. Kernighan and P. J. Plauger, *The Elemenets Of Programing Style*. Estados Unidos: McGraw-Hill Book Company, 1978.
- [27] A. Hunt and D. Thomas, *The Pragmatic Programmer: From Journeyman to Master*. The Pragmatic Bookshelf, 1999.
- [28] G. Yzaguirre, "Manifiesto del Laboratorio de Software Libre." 2016. Available: https://labsl.multimediales.com.ar/Manifiesto_del_Laboratorio_de_Software_Libre_.html
- [29] M. Good, "MusicXML: An Internet-Friendly Format for Sheet Music," *Proceedings of XML*, 2001, Available: <http://michaelgood.info/publications/music/musicxml-an-internet-friendly-format-for-sheet-music/>
- [30] J. Leek, "The future of education is plain text." 2017. Available: <https://simplystatistics.org/2017/06/13/the-future-of-education-is-plain-text>
- [31] (n.d.), "¿Que es el Software Libre?" 2001. Available: <https://www.gnu.org/philosophy/free-sw.es.html>
- [32] B. Moolenaar, "Seven habits of effective text editing." 2000. Available: <http://moolenaar.net/habits.html>
- [33] J. Steyn, "Music Markup Language." 2001. Available: <https://steyn.pro/mml>

- [34] C. Clark and A. Tindale, “Flocking: A Framework for Declarative Music-Making on the Web,” *The Joint Proceedings of the ICMC and SMC*, vol. 1, no. 1, pp. 50–57, 2014.
- [35] E. S. Raymond, *The Art of UNIX Programming*. Estados Unidos: Addison-Wesley Professional, 1999.
- [36] P. Graham, “Beating the Averages,” 2001, p. 01. Available: <http://www.paulgraham.com/avg.html>
- [37] E. S. Raymond, “The Cathedral and the Bazaar,” 1997, pp. 01–99. doi: [10.1007/s12130-999-1026-0](https://doi.org/10.1007/s12130-999-1026-0).
- [38] D. Grela, “Análisis Musical: Una Propuesta Metodológica,” 1992, pp. 04–05.
- [39] R. A. Penfold, *Advanced MIDI Users Guide*. United Kingdom: PC Publishing, 1992.
- [40] D. Neil, *Practical Vim*. Dallas, Texas - Raleigh, North Carolina: The Pragmatic Bookshelf, 2012.
- [41] P. Graham, *Hackers and Painters= Big Ideas from the Computer Age*. Estados Unidos: O’Reilly Media, 2004.
- [42] F. Lerdahl and R. Jackendof, *A Generative Theory of Tonal Music*. Estados Unidos: The MIT Press, 1996.
- [43] O. Ben-Kiki, C. Evans, and B. Ingerson, “Yaml ain’t markup language (yaml™) version 1.1,” *yaml. org, Tech. Rep*, p. 23, 2005.
- [44] E. C. Ben-Kiki Oren and Ingy, “YAML Version 1.2 Specification.” 2009. Available: <http://yaml.org/spec/1.2/spec.html>
- [45] L. Wall, “Perl, the first postmodern computer language.” 1999. Available: <https://www.perl.com/pub/1999/03/pm.html/>
- [46] G. V. Rossum, “Python 3.7.” 2018. Available: <https://docs.python.org/3/>
- [47] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- [48] (n.d.), “PyYAML is a full-featured YAML framework for the Python programming language.” 2018. Available: <https://pyyaml.org/>
- [49] M. S. Cuthbert, “music21: A toolkit for computer-aided musicology.” 2018. Available: <http://web.mit.edu/music21>
- [50] (n.d.), “The Python Standard Library.” 2018. Available: <https://docs.python.org/3/library/index.html>
- [51] B. Moolenaar, “VIM.” 2018. Available: <https://www.vim.org/docs.php>
- [52] L. Torvalds, “GIT.” 2018. Available: <https://git-scm.com/docs>
- [53] L. Torvalds and J. Hamano, “Git: Fast version control system,” *git-scm.com*, 2010, Available: <http://git-scm.com>

- [54] G. Brandl and the S. team, “Python Documentation Generator.” 2018. Available: <https://sphinx-doc.org/en/master>
- [55] J. F. Allen, “Maintaining knowledge about temporal intervals,” *Communications of the ACM.*, pp. 832–843, 1983, doi: [10.1145/182.358434](https://doi.org/10.1145/182.358434).
- [56] P. Schaeffer, *Tratado de los objetos musicales*. 1966.
- [57] L. M. Samaruga, “Un modelo de representación y análisis estructural de la música electroacústica .” {PhD} {Thesis}, Universidad Nacional de Quilmes, 2016.
- [58] Y. Franco, “Tesis de Investigación.” 2011. Available: <http://tesisdeinvestig.blogspot.com/2011/06/entrevistas.html>
- [59] K. Shaffer, “Make Stunning Schenker Graphs with GNU Lilypond.” 2005. Available: <https://www.linuxjournal.com/article/8364>
- [60] M. C. Wirts, “MIDIUtil.” 2016. Available: <https://midiutil.readthedocs.io>
- [61] J. Variego, *Composición algorítmica. Matemáticas y ciencias de la computación en la creación musical*. Universidad Nacional de Quilmes, 2018.
- [62] S. T. Pope, “Music Notations and the Representation of Musical Structure and Knowledge.” *Perspectives of New Music*, vol. 24, no. 2, pp. 156–189, 1986, doi: [10.2307/833219](https://doi.org/10.2307/833219).
- [63] M. M. Association (MMA), “Standard MIDI Files (SMF) Specification.” 1996. Available: <https://www.midi.org/specifications-old/item/standard-midi-files-smf>
- [64] G. Huntley, “NoYAML.” 2019. Available: <https://noyaml.com/>
- [65] G. Huntley, “Interprete de Comandos.” 2019. Available: <https://noyaml.com/>
- [66] (n.d.), “Lenguaje de Marcado.” 2019. Available: https://es.wikipedia.org/wiki/Lenguaje_de_marcado
- [67] (n.d.), “Interprete de Comandos.” 2019. Available: [https://es.wikipedia.org/wiki/Interprete_\(informática\)](https://es.wikipedia.org/wiki/Interprete_(informática))
- [68] (n.d.), “YAML Test Matrix.” 2019. Available: <https://matrix.yaml.io/valid.html>
- [69] (n.d.), “Lenguaje específico de dominio.” 2019. Available: https://es.wikipedia.org/wiki/Lenguaje_específico_de_dominio
- [70] (n.d.), “Analizador sintáctico.” 2019. Available: https://es.wikipedia.org/wiki/Analizador_sintáctico
- [71] (n.d.), “Interprete de Comandos.” 2019. Available: https://es.wikipedia.org/wiki/Codificación_digital
- [72] G. Haus and L. Ludovico, “Music Representation of Score, Sound, MIDI, Structure and Metadata All Integrated in a Single Multilayer Environment Based on XML,” 2007. doi: [10.4018/9781599046631.ch014](https://doi.org/10.4018/9781599046631.ch014).

- [73] G. Godlewski, “Anthropology of the Word: The Stepsister of Linguistic Anthropology,” *Anthropological Journal of European Cultures*, vol. 24, no. 1, pp. 7–23, 2015.
- [74] S. Oualline, *Vi iMproved*. New Riders Publishing, 2001.