

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

Resumen

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 de académica con Pandoc | 1 |
| 1.1 Pandoc | 2 |
| 2 Metodología | 3 |
| 2.1 Integración | 3 |
| 2.2 Gráficos y diagramas | 3 |
| 2.3 Bibliografías sofisticadas | 6 |
| 2.4 Notación matemática | 7 |
| 3 Resultados | 8 |
| 3.1 Sintaxis extendida de Markdown | 8 |
| 3.2 Numeración y referencias cruzadas | 8 |
| 4 Conclusión | 9 |
| 4.1 Objetivos | 9 |
| 4.2 Aporte | 9 |
| 4.3 Limitaciones | 10 |
| 4.4 Futuras líneas de trabajo | 10 |
| Referencias | 11 |

1 Producción de académica con Pandoc

Este proyecto describe la confección de escritos académicos o de complejidad considerable, sin la necesidad de interfaces gráficas. Evitar el uso de formatos codificados o de poca legibilidad beneficia a todos los usuarios, que deben poder encontrar lo que necesitan, comprender lo que encuentran y usarlo para realizar tareas [1].

El objetivo de este trabajo es un entorno de autoría de textos en el cual *Pandoc* es la pieza central que actúa como 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 [2, 3].

Mediante integraciones sencillas se consigue una infraestructura robusta con funciones diseñadas para gestionar exposición de extensas referencias, múltiples citas y referencias a diferentes fuentes, notación matemática, generación de gráficos y diagramas, entre otras capacidades avanzadas, necesarias para la producción de documentación técnica y científica, todo el proceso es controlado mediante línea de comandos sin depender de interfaces captivas promoviendo la transparencia, la claridad y la reproducción [4, pp. 88-97].

1.1 Pandoc

Pandoc es una biblioteca de *Haskell* 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 formatos y procesar textos [5].

El diseño de *Pandoc* es modular, esta conformado por un conjunto de lectores, que analizan el texto en un formato determinado y producen una representación nativa del documento en un árbol de sintaxis abstracta (Abstract Syntax Tree - AST) y un conjunto de escritores, que convierten esta representación a un formato de destino [6, 7].

1.1.1 Interfaz

La principal característica de las herramientas empleadas este proyecto es que están preparadas para recibir instrucciones alojadas en archivos de texto. Entre los beneficios que trabajar de este modo habilita se destacan cuestiones de accesibilidad y la posibilidad de gestionar el conocimiento de la misma manera que se produce software [8–10].

Separar contenido, referencias, estilos y procesos, en un contexto de organizaciones con actividades relacionadas a la publicación, donde la complejidad no solo reside en los documentos sino que también en la tarea, dado que involucra a múltiples agentes (como autores, correctores y editores, entre otros); libera al autor de problemas estéticos y devuelve el control de estilo a la organización garantizando unidad en estética en la composiciones gráfica resultante de diversos productos.

Esta formación en los estudiantes introducirá en la fuerza de trabajo una nueva capacidad con una inclinación arraigada y fundamental hacia la investigación reproducible [11]. Además de la funcionalidad que se señala como futura línea de trabajo, la posibilidad de generar documentos como el presente mediante operaciones remotas automáticas.

1.1.2 Markdown

Markdown es una sintaxis de formato de texto plano. 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, pesos tamaño, y características especiales (como hipervínculos). Al texto resultante se le conoce como texto formateado, texto con estilos, o texto enriquecido [12].

Lo que distingue a *Markdown* de muchas otras sintaxis de marcado ligero, es su énfasis en la legibilidad. El objetivo principal del diseño de la sintaxis de formato de *Markdown* es hacerla lo más legible posible. La idea es que un documento con formato *Markdown* sea publicable tal cual, como texto plano, sin que parezca que ha sido marcado con etiquetas o instrucciones de formato.

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.

2 Metodología

En este capítulo se describe el método propuesto y utilizado para producir el presente documento.

Primero se introduce a la integración de diferentes herramientas, algunas distribuidas junto con *Pandoc* y otras aportes independientes de la comunidad. Luego sistema de diagramación y generación gráficos que permite crear visualizaciones utilizando texto y código. Seguido se expone el sistema citas y referencias bibliográficas. Para concluir este capítulo se exponen cuestiones relacionadas a la notación matemática.

2.1 Integración

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

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.

2.2 Gráficos y diagramas

La diagramación y conllevan tiempo a los investigadores y desarrolladores y quedan obsoletas rápidamente. Pero no tener diagramas o documentación arruina la productividad y perjudica el aprendizaje de la organización.

Se destina esta tarea a *pandoc-plot* es un filtro de *Pandoc* para generar figuras a partir de bloques de código en documentos. Los gráficos a continuación están generados a partir del código incluido en el fichero *Markdown* original utilizando *pandoc-plot*, para que demostren las posibilidades de esta herramienta.

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

```

import numpy as np
import matplotlib.pyplot as plt

theta = np.arange(0, 2 * np.pi, .01)[1:]
r = theta - np.pi
positive_r = r >= 0

fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(10, 5), subplot_kw={'polar': True})

for ax in (ax1, ax2):
    if ax == ax2:
        # change negative r values to positive, rotating theta by 180°
        theta = np.where(r >= 0, theta, theta + np.pi)
        r = np.abs(r)
        ax.plot(theta[positive_r], r[positive_r], color='skyblue')
        ax.plot(theta[~positive_r], r[~positive_r], color='tomato')
    ax1.set_title('Default: negative $r$ \non same side as $\theta$')
    ax2.set_title('Negative $r$ on other side')

import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(
    subplot_kw = {'projection': 'polar'}
)
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.title('This is an example figure')

import numpy as np
import matplotlib.pyplot as plt

np.random.seed(23)

# Compute areas and colors
N = 150
r = 2 * np.random.rand(N)
theta = 2 * np.pi * np.random.rand(N)
area = 200 * r**2
colors = theta

fig = plt.figure()
ax = fig.add_subplot(111, projection='polar')
c = ax.scatter(theta, r, c=colors, s=area, cmap='hsv', alpha=0.75)

```

```

plt.title('This is an example figure')

@startuml
!theme plain

package "Some Group" {
    HTTP - [First Component]
    [Another Component]
}

node "Other Groups" {
    FTP - [Second Component]
    [First Component] --> FTP
}

cloud {
    [Example 1]
}

database "MySQL" {
    folder "This is my folder" {
        [Folder 3]
    }
    frame "Foo" {
        [Frame 4]
    }
}

[Another Component] --> [Example 1]
[Example 1] --> [Folder 3]
'[Folder 3] --> [Frame 4]

@enduml

@startuml
robust "DNS Resolver" as DNS
robust "Web Browser" as WB
concise "Web User" as WU

@0
WU is Idle
WB is Idle
DNS is Idle

@+100

```

```

WU -> WB : URL
WU is Waiting
WB is Processing

@+200
WB is Waiting
WB -> DNS@+50 : Resolve URL

@+100
DNS is Processing

@+300
DNS is Idle
@enduml

```

Alice: Great! ~~~ —>

2.3 Bibliografías sofisticadas

Para la gestión de bibliografía. Por ejemplo, esto significa que puede escribir una referencia como `@moolenaar2000` o también `[@knuth1986texbook p.3-9]` y *Pandoc* la 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.

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

2.3.1 BibLaTeX

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.

2.3.2 CLS

Las referencias son una pieza clave 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 no sólo ayuda 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, el Lenguaje de Estilos de Citación (Citation Style Language - CSL) es el descriptor utilizado [13].

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

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.

2.4 Notación matemática

MathJax

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.

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 resultado de este proyecto es la integración de diferentes piezas de software y andamiaje necesario para reproducir este proyecto: configuraciones, estructura, filtros, plantillas (*LaTeX*, CLSs, resaltado de sintaxis) y ejemplo de operaciones remotas automáticas.

Para recrear el proceso que se utilizó para confeccionar este documento. Un ejemplo de cómo usar esta herramienta es el mismo comando que sirve para conseguir este documento desde su fuente en *Markdown*.¹

```
$ 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 un sistema con una instancia de *Pandoc* funcionando produce 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 de la tabla de contenidos, y archivo de entrada y de salida.

3.1 Sintaxis extendida de Markdown

Hay un aspecto en el que los objetivos de *Pandoc* difieren de los originales de Markdown. Mientras que *Markdown* fue diseñado para la generación de HTML en mente, *Pandoc* está preparado para producir múltiples formatos de salida.

La versión mejorada de *Markdown* de *Pandoc* entiende una versión ampliada y ligeramente revisada de la sintaxis original de Gruber, incluye sintaxis para tablas, listas de definiciones, bloques de metadatos, notas a pie de página, citas, matemáticas y mucho más.

<https://github.com/gfm/> <https://pandoc.org/MANUAL.html#pandocs-markdown>

3.2 Numeración y referencias cruzadas

<https://github.com/lierdakil/pandoc-crossref/raw/master/docs/demo/output.pdf>

¹Como conseguir una instalación funcional de esta fuera de los alcances de este artículo.

4 Conclusión

Este capítulo concluye el estudio. El objetivo de es investigar el proceso de producción de documentos con herramientas rudimentales. En primer lugar, se responde a las preguntas de investigación. El segundo subcapítulo presenta la contribución de esta tesis, y los dos últimos subcapítulos presentan las limitaciones del estudio y las sugerencias para desarrollos futuros, respectivamente.

4.1 Objetivos

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 predefinidos por la comunidad, etiquetas de bibliografía altamente personalizable 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 en producción y basada en macros sin cambiar las fuentes de datos.

4.2 Aporte

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

Si bien que este proyecto está enfocado a la producción de literatura académica, esta misma cadena puede ser considerada en en el desarrollo de cualquier sistemas 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 descripto de respaldo del la información en contenedores de formato simple y legible, sin codificar u opacar habilitando ser manipulados con herramientas obicuas, con

Pero en una implementación organizacional esto puede ser solucionado ejecutando en servidor públicos donde con acceso web 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 o

Aunque los escuadrones son autónomos, es importante que los especialistas (por ejemplo, los desarrolladores de) se alineen en las mejores prácticas.

4.3 Limitaciones

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*.

Los módulos empleados maduraron y se incluyen en la la distribución principal de *Pandoc* cambiar configuraciones agregar funcionalidades y rutinas puede variar y depender de versiones de las herramientas en las diferente distribuciones y sistemas operativos puede inferir en la integración y operaciones.

4.4 Futuras líneas de trabajo

Se señala como áreas de desarrollo

4.4.1 Operaciones remotas automáticas

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

disponible el repositorio con



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

4.4.2 Revisión sistemática de literatura

Para revisiones de literatura que comprometen múltiples bibliografías y [14]

Que deben 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

Referencias

- [1] W. Caleb McDaniel, «Why (and how) I wrote my academic book in plain text», *W. Caleb McDaniel*. Disponible en: <http://wcaleb.org/blog/my-academic-book-in-plain-text>
- [2] John MacFarlane, «Pandoc - a universal document converter», *Pandoc - a universal document converter*. 2022. Accedido: 14 de septiembre de 2022. [En línea]. Disponible en: <https://pandoc.org/>
- [3] D. E. Knuth, D. Knuth, y D. Bibby, *The TeXbook*. Addison-Wesley, 1986. Disponible en: www-cs-faculty.stanford.edu/~knuth/abcde.html
- [4] M. Gancarz, *Linux and the Unix Philosophy*. Elsevier Science, 2003. Disponible en: <https://books.google.com.ar/books?id=qqstCSlk5MIC>
- [5] S. Marlow *et al.*, «Haskell 2010 language report», 2010. <http://www.haskell.org>
- [6] J. Jones, «Abstract Syntax Tree Implementation Idioms», *Pattern Languages of Program Design*, 2003, Disponible en: <http://hillside.net/plop/plop2003/Papers/Jones-ImplementingASTs.pdf>
- [7] I. Neamtiu y I. Bind, «Understanding source code evolution using abstract syntax tree matching», 2005, pp. 2-6.
- [8] A. Hunt y D. Thomas, *The Pragmatic Programmer: From Journeyman to Master*. Pearson Education, 1999. Disponible en: <https://books.google.com.ar/books?id=5wBQEp6ruIAC>
- [9] D. A. S. U. Harvard, «Use plain language», *Digital Accessibility*. Digital Accessibility Services. Disponible en: accessibility.huit.harvard.edu/use-plain-language
- [10] B. Moolenaar, «Seven habits of effective text editing», 2000. mooleenaar.net/habits.html
- [11] B. Baumer y D. Udwin, «R Markdown», *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 7. Wiley, pp. 167-177, febrero de 2015. doi: [10.1002/wics.1348](https://doi.org/10.1002/wics.1348).
- [12] J. Gruber, «Markdown: Syntax», *Daring Fireball: Markdown Syntax Documentation*. Disponible en: <https://daringfireball.net/projects/markdown/syntax#philosophy>
- [13] R. M. Zelle, F. G. Bennet Jr, y B. D'Arcus, «Citation style language 1.0. 1: Language specification, 2012», *URL: http://citationstyles.org/downloads/specification.html*.
- [14] B. Kitchenham, «Evidence-based software engineering and systematic literature reviews», en *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2006, vol. 4034 LNCS, p. 3. doi: [10.1007/11767718_3](https://doi.org/10.1007/11767718_3).