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 2022

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

Contenidos

1 Producción académica con Pandoc

Este proyecto propone la confección de escritos académicos o de complejidad considerable, sin la necesidad de interfaces gráficas. Promover el uso formatos no codificados o de alta legibilidad beneficia a todos los usuarios, que deben poder encontrar facilmente lo que necesitan, comprender lo que encuentran y usarlo para realizar tareas [1].

El objetivo de este trabajo es un entorno de autoría 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 biblografías, múltiples citas y referencias a diferentes fuentes, notación matemática, generación gráficos y diagramas, entre otras capacidades avanzadas, necesarias en la producción de documentación técnica y científica, todo el proceso es controlado mediante linea de comandos sin depender de interfaces captivas, promoviendo la transparencia, claridad y reproducción [4, pp. 88-97].

1.1 Interfaz de usuario textual

La principal característica de las herramientas y formatos involucrados en este proyecto es que están preparadas para interpretar instrucciones textuales. De los beneficios que trabajar de este modo habilita se destacan cuestiones de accesibilidad y la posibilidad de gestionar la exposición de conocimiento de la misma manera que se produce software [5–7].

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 (autores, correctores y editores, entre otros) y devuelve el control de estilo a la organización, garantizando unidad en estética en la composición gráfica resultante de diversos productos.

Esta formación introducirá en la fuerza de trabajo una nueva capacidad con una inclinación arraigada y fundamental hacia la investigación reproducible [8]. El lenguaje sigue siendo la mejor interfaz que se ha utilizado. Es sencillo, componible y ubicuo, está disponible en todos los sistemas. Es fácil de mantener, automatizar y ampliar [9].

1.2 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 [10].

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 [11, 12].

1.3 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 [13].

Lo que distingue a *Markdown* de muchas otras sintaxis de marcado ligero, es su énfasis en la legibilidad. El objetivo prinsipal 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 Metodotología

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

Primero se describe la integración de diferentes piezas de software, algunas distribuidas junto con *Pandoc* y otras aportes independientes de la comunidad. Seguido se presenta sistema de diagramación y generación gráficos que permite crear visualizaciones utilizando texto y código. Luego se expone el sistema citas y referencias bibliográficas. Para concluir este capitulo 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 Sintactic Three - AST), y un conjunto de registros, que convierten esta representación nativa en un formato de destino.

Ademas, incluye un potente sistema para escribir filtros, para incluir un formato de entrada o de salida basta con añadir un lector o un escritor. También es posible crear filtros personalizados para modificar el AST intermedio.

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, un potente sistema de citas y bibliografías automáticas y la generación de gráficos mediante código.

2.2 Gráficos y diagramas

La diagramación conlleva tiempo a los investigadores y desarrolladores, los gráficos producidos suelen quedar obsoletos rápidamente. Pero no tener diagramas o documentación arruina la productividad y perjudica el aprendizaje de la organización.

Se destina esta tarea a *pandoc-plot*, un filtro de *Pandoc* para generar figuras a partir de bloques de código en los documentos [14]. Al actual, *pandoc-plot* es compatible con el siguiente conjunto de herramientas de trazado: matplotlib; *plotly_python, plotly_r, matlabplot, mathplot, octaveplot, ggplot2, gnuplot, graphviz, bokeh, plotsjl* y *plantuml*.

En este trabajo se implementan dos de ellas, *Matplotlib* y *PlantUML* [15, 16]. En los apartados a continuación se exponen gráficos generados con dichas herramientas partir del condigo incluido en el fichero *Markdown* original, para que demostrar las posibilidades de esta herramienta.

2.2.1 Matplotlib

```
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 180º
    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')
```

2.2.2 PlantUML

```
@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
```

2.3 Citas, referencias y bibliografía

Para citar, enlazar a referencias y exposición de bibliografía consultada se emplea *BibLaTeX*, una herramienta y un formato de archivo que se utilizan para describir y procesar listas de referencias, sobre todo en combinación con documentos *LaTeX*.

Los datos bibliográficos de entrada 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 una reimplementación completa de las facilidades bibliográficas proporcionadas por *LaTeX*. Esto significa, por ejemplo que al declarar una referencia como `@moolenaar2000` o también `[@knuth1986texbook p.3-9]` *Pandoc* las 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 a la referencia bibliografía con el formato adecuado al final del documento.

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. *BibLaTeX* tiene muchas características que rivalizan o superan a otros sistemas bibliográficos.

2.3.2 Lenguaje de Estilo de Citación

La 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 Estilo de Citación (Citation Style Languaje - CSL) es el descriptor utilizado es un formato basado en XML para describir el formato de citas, notas y bibliografías [17].

2.3.3 Pandoc crossref

pandoc-crossref es un filtro de para numerar figuras, ecuaciones, tablas y referencias cruzadas a las mismas [18]. En Apéndice B sec. **??**, se expone el documento oficial de demostración las capacidades de esta herramienta, incluido en la cadena de procesos de estos proyectos.

2.4 Notación matemática

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 incluyendo sintaxis *MathJax* y la traducción a *MathML*.

Cuando $a \neq 0$, hay dos soluciones a (axˆ2 + bx + c = 0) las cuales son

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

Transformación de contenidos: 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 fuentes SVG, en lugar de de imágenes de mapa de bits, por lo que las ecuaciones se escalan con el texto circundante.

Modular la entrada y la salida: MathJax es altamente modular en la entrada y la salida. Utiliza MathML, TeX, y ASCIImath como entrada y 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: fuentes de entrada, configuraciones, estructura, filtros, plantillas (*LaTeX*, CLSs, resaltado de sintaxis) y un ejemplo flujo de trabajo acciones integración remota automatizada.

Para recrear este proceso, principalmente hay 2 opciones:

La mas directa es realizar un *fork* el repositorio en el cual esta alojado el contenido en linea [20]. Después de realizar modificaciones necesarias, esto dispara acciones en el repositorio y genera este documento.

Para trabajar en una copia local es necesario es ejecutar los siguientes comando en un terminal de sistema para, clonar el contenido, inicializar el proyecto y generar el documento. [1].

```
$ git clone https://github.com/lifofernandez/article-boilerplate.git
$ cd article-boilerplate
$ sudo make install
$ pandoc README.md \
    -F pandoc-plot --metadata-file=metadata.yaml --mathjax \
    -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
```

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.

En Apéndice A (Sec. **??**) expone la versión mejorada de *Markdown* de *Pandoc* que comprende una versión ampliada y ligeramente revisada de la sintaxis original[2].

[1] Conseguir una instalación funcional de *pandoc* y sus dependencias es condicionante el sistema en el que se ejecute. Para instrucciones especificas consultar las indicaciones su autor [21].

[2] El contenido de los apéndices se encuentran en su idioma original.

Incluye sintaxis para tablas, listas de definiciones, bloques de metadatos, notas a pie de página, citas y matemáticas y entre otros [22].

3.2 Numeración y referencias cruzadas

Para consultar una lista completa de las funcionalidades avanzadas de *pandoc-crossref* el módulo de *pandoc* para realizar referencias cruzadas. Acompaña este artículo la demostración de su autor en Apéndice B (Sec. **??**).

4 Conclusión

Este capítulo concluye el estudio. En primer lugar, se cubren los objetivos de investigación. El segundo subcapítulo presenta la contribución de esta trabajo, y los dos últimos subapartados presentan las limitaciones del estudio y las sugerencias para desarrollos futuros, respectivamente.

4.1 Alcance

El animo de este proyecto es desarrollar una cadena de producción de documentos científicos y técnicos sin depender de interfaces gráficas o captivas.

Las características generales de este entorno son: formatos libres y abiertos, componentes aislados, compactos y robustos; amplia compatibilidad con requisitos de estilo, predefinidos por la comunidad o personalizados por el usuario. Vinculación a fuentes de datos remotas para publicaciones recurrentes con información dinámica.

4.2 Aporte

Es intención que este trabajo que sirva como punto de partida en contextos similares, reutilizado patrones de diseño y siguiendo guía de buenas prácticas en la producción de documentos gráficos de alta complejidad.

Si bien este proyecto está enfocado a la producción de literatura académica, esta misma cadena de producción puede ser aplicada en el desarrollo de cualquier otro sistema como por ejemplo, gestión documental, registros médicos, documentos legales, certificados legales, entre otros.

En una implementación organizacional esto puede ser aprovechado ejecutando en servidor remotos como servicio de preparación de documentos gráficos. En aquellos contextos que los productos gráficos se generan mediante rutinas directamente de bases de datos, una capa codificada extra que opaca la relación entre el interprete y el contenido, se recomienda un proceso similar al descripto de respaldo de la información en contenedores de formato simple y legible, sin codificar.

Aunque los escuadrones sean autónomos, es importante que los especialistas (por ejemplo, editores) 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 pueden tener diferencias.

Pandoc intenta conservar los elementos estructurales de un documento, pero no los detalles de formato, como el tamaño de los márgenes. Algunos elementos del documento, como por ejemplo tablas complejas, pueden no encajar en el modelo de documento simple de *Pandoc*.

4.4 Futuras lineas de trabajo

Se señala como áreas de desarrollo

4.4.1 Entrega continua

Como se puede comprar en el respositorio que aloja el este proyecto el documento PDF de salida puede ser producido
mediante Operaciones remotas automáticas [23].

Servicios como estos acortan las brecha entre las actividades y los equipos de producciòn, al imponer la automatización en la construcción y entrega de documentos. Los servicios de entrega continuna compilan los cambios incrementales en el contenido de los autores, los enlazan, los empaquetan y los ejectuan en un entorno remoto preconfigurado.

4.4.2 Revisión sistemática de literatura

Este proceder promueve capacidades como ordenación personalizable, Bibligrafías jeraquizadas por sección; soporte de poliglosia para el cambio automático de idioma de las entradas y citas bibliográficas; modelo de datos personalizable para que los usuarios puedan definir sus propios tipos de datos bibliográficos; validación de datos bibliográficos con respecto a un modelo.

En investigaciones del tipo revisiones de literatura, donde se involucran múltiples cuerpos bibliográocos con diferentes ordenación y modos exponerse, enfoques como este pueden simplificar el proceso [24].

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] J. 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] 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

[6] D. A. S. U. Harvard, «Use plain language», *Digital Accessibility*. Digital Accessibility Services. Disponible en: accessibility.huit.harvard.edu/use-plain-language

[7] B. Moolenaar, «Seven habits of effective text editing». moolenaar.net, 2000. Disponible en: moolenaar.net/habits.html

[8] 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.

[9] R. Scape, «Text Is the Universal Interface». 2022. Disponible en: https://scale.com/blog/text-universal-interface

[10] S. Marlow *et al.*, «Haskell 2010 language report». 2010. Disponible en: http://www.haskell.org

[11] J. Jones, «Abstract Syntax Tree Implementation Idioms», *Pattern Languages of Program Design*, 2003, Disponible en: http://hillside.net/plop/plop2003/Papers/Jones-ImplementingASTs.pdf

[12] I. Neamtiu y I. Bind, «Understanding source code evolution using abstract syntax tree matching», 2005, pp. 2-6.

[13] J. Gruber, «Markdown: Syntax», *Daring Fireball: Markdown Syntax Documentation*. Disponible en: https://daringfireball.net/projects/markdown/syntax#philosophy

[14] L. R. de Cotret, «Pandoc Plot». 2019. Disponible en: https://laurentrdc.github.io/pandoc-plot

[15] D. F. Hunter John; Dale y T. M. development team, «Matplotlib: Visualization with Python». 2013. Disponible en: https://matplotlib.org/

[16] A. Roques, «PlantUML Generate UML diagram from textual description». 2013. Disponible en: https://plantuml.com/

[17] 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*.

[18] N. Yakimov, «pandoc-crossref filter». 2013. Disponible en: https://lierdakil.github.io/pandoc-crossref/

[19] S. F. Conservancy, «Contributing to a Project». 2022. Disponible en: https://git-scm.com/book/en/v2/GitHub-Contributing-to-a-Project

[20] L. Fernández, «Article Boilerplate». 2022. Disponible en: https://github.com/lifofernandez/article-boilerplate

[21] J. MacFarlane, «Installing Pandoc». 2022. Disponible en: https://pandoc.org/installing.html

[22] J. MacFarlane, «Pandoc's Markdown». 2022. Disponible en: https://pandoc.org/MANUAL.html#pandocs-markdown

[23] J. MacFarlane, «Github Actions». 2022. Disponible en: https://pandoc.org/installing.html#github-actions

[24] 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.