

# Simple PDF

---

Martín Goñi

2/2/2026

# ¿Que es SPF?

Simple PDF, es un DSL para generar PDFs, es decir un lenguaje que al ser interpretado produce PDFs. Es más este informe esta hecho con **SPF**.

## Sobre SPF

### Consideraciones de diseño

A la hora de diseñar **SPF** me inspire en **LaTeX**, pero sin mantenerme atado a sus convenciones o decisiones de diseño, lo trate como una fuente de inspiracion. En particular tome otra direccion en cuanto a la forma de los comandos, permitiendo solo un argumento; Al mismo tiempo las opciones deben tener asociada una clave indicando a que corresponde el valor. Finalmente simplifique la forma en la que estructura el documento, ya no es necesario separar el titulo y documento de forma explicita.

Los comandos implementados son un subconjunto de los disponibles en **LaTeX**, elegidos para permitir generar documentos (mayormente)completos, manteniendo al mismo tiempo la cantidad total de comandos baja.

## Sintaxis

Para representar la sintaxis se usan ciertas convenciones; El simbolo \* indica cero o mas ocurrencias de lo que encierre mientras que el + indica una o mas ocurrencias. Los comentarios no estan incluidos pues no son parte de la sintaxis en si, son añadidos en el parser para la conveniencia del usuario. A continuación se presenta la sintaxis concreta de **SPF**:

```
digit      ::= '0' | '1' | ... | '9'
integer    ::= <digit> (<digit>)*
float      ::= <integer> '.' <integer>
letter     ::= 'a' | ... | 'z' | 'A' | ... | 'Z'
letterStr   ::= <letter> (<letter>)*
specialChars ::= '\' | '{', '}' | '[' | ']' | '"' | '/' | '|'
nonSpecialChars ::= c / c != <specialChars>
filepath    ::= (<letterStr> | '/' | '\' | '-' | '_' | '.' | '+')+

#| Opciones
optionList   ::= '[' optionListElems ']'
optionListElems ::= <optionPair> (',' <optionPair>)*
optionPair    ::= <letterStr> ':' <optionValue>
optionValue   ::= <boolValue>
             | <numberValue>
             | <literalValue>
             | <identifierValue>

boolValue    ::= 'true' | 'false'
numberValue   ::= <integer> | <float>
literalValue  ::= ''' <nonSpecialChars> '''
identifierValue ::= <letterStr>

#| Texto
text        ::= ('\\bold' | '\\italic' | '\\emph') '{' <textChars> '}'
textChars   ::= (<nonSpecialChars>)+ <textChars> | '\\' <specialChars> <textChars> |
<empty>

#| Configuracion del documento
config      ::= '\\config{ ' <configName> ' }' <optionList>
configName   ::= 'size' | 'pagenumbering' | ... | 'verbatimnumbering'

#| Metadata del documento
```

```

metadata    ::= <title> <autho> <date>
title       ::= '\title{' <text> '}' | <empty>
author      ::= '\author{' <text> '}' | <empty>
date        ::= '\date{' <text> '}' | <empty>

#| Comandos
command    ::= <text>
            | <paragraph>
            | <section>
            | <subsection>
            | <figure>
            | <table>
            | <list>
            | <verbatim>
            | <newpage>
            | <hline>

paragraph   ::= '\begin{paragraph}' <optionList> <text> '\end{paragraph}'
section     ::= '\section{' <text> '}' <optionList>
subsection   ::= '\subsection{' <text> '}' <optionList>
figure      ::= '\figure{' <filepath> '}' <optionList>
table        ::= '\begin{table}' <optionList> <tableContents> '\end{table}'
tableContents ::= <text> ('|' <text>)* '\break'
list         ::= '\begin{list}' <optionList> <listContents> '\end{list}'
listContents  ::= '\item' <text>
verbatim     ::= '\begin{verbatim}' <optionList> <verbatimContents>
                '\end{verbatim}'
verbatimContents ::= (<anyChar>)+ <notFollowedBy '\end'>
newpage      ::= '\newpage'
hline        ::= '\hline' <optionList>

#| Documento
document    ::= (<config>)* <metadata> (<command>)+
```

Hay una aclaracion importante sobre la sintaxis para **verbatim**, esto es que acepta cualquier caracter hasta encontrar "*\end*".