

# Una Introducción Interactiva a $\text{\LaTeX}$

## Parte 4: Adaptando el documento a nuestras necesidades

Luis A. Guanuco

18 de mayo de 2015



## Estilo De Las Páginas: Nativos

- ▶  $\text{\LaTeX}$  soporta diferentes combinaciones de cabeceras y pies de páginas. `\pagestyle` define cuál emplearse.
  - ▶ `empty`
  - ▶ `plain`
  - ▶ `headings`
  - ▶ `myheadings`
- ▶ Es posible cambiar el estilo de la página actual con la orden pies de páginas `\thispagestyle`.

# Estilo De Las Páginas: Personalizados

El estilo `myheadings` permite modificar el contenido de la cabecera.

```
\documentclass[a5paper]{report}
\usepackage{kantlipsum}
\pagestyle{myheadings}
\markright{Nombre y Apellido}
\title{Informe}
\author{Alumno}
\begin{document}

\maketitle

\chapter{Introducci'on}

\kant[1-6]

\section{Antecedentes}
\kant[6-10]

\end{document}
```

*Nombre y Apellido*

5

pure reason.

## 1.1 Antecedentes

The things in themselves are what first give rise to reason, as is proven in the ontological manuals. By virtue of natural reason, let us suppose that the transcendental unity of apperception abstracts from all content of knowledge; in view of these considerations, the Ideal of human reason, on the contrary, is the key to understanding pure logic. Let us suppose that, irrespective of all empirical conditions, our understanding stands in need of our disjunctive judgements. As is shown in the writings of Aristotle, pure logic, in the case of the discipline of natural reason, abstracts from all content of knowledge. Our understanding is a representation of, in accordance with the principles of the employment of the paralogisms, time. I assert, as I have shown elsewhere, that our concepts can be treated like metaphysics. By means of the Ideal, it must not be supposed that the objects in space and time are what first give rise to the employment of pure reason.

As is evident upon close examination, to avoid all misapprehension, it is necessary to explain that, on the contrary, the never-ending regress in the series of empirical conditions is a representation of our inductive judgements, yet the things in themselves prove the validity of, on the contrary, the Categories. It remains a mystery why, indeed, the never-ending regress in the series of empirical conditions exists in philosophy, but the employment of the Antinomies, in respect of the intelligible charac-

# Estilo De Las Páginas: Personalizados

El paquete `fancyhdr` provee comandos para definir el contenido del lado izquierdo, centro y derecho, tanto del encabezado como el pie de página.

```
\documentclass[a5paper]{report}
\usepackage{kantlipsum,graphicx,lastpage}
\usepackage{fancyhdr}
\pagestyle{fancy}

\title{Informe}
\author{Alumno}
\lhead{Nombre y Apellido}
\chead{\includegraphics[width=0.02\textwidth]{images/logoUTN}}
\rhead{\today}
\lfoot{Prof: Nombre y Apellido}
\cfoot{\thepage de \pageref{LastPage}}
\rfoot{Universidad -- Facultad}
\renewcommand{\headrulewidth}{0.4pt}
\renewcommand{\footrulewidth}{0.4pt}

\begin{document}
\maketitle
\chapter{Introducci'on}
\kant[1-6]

\section{Antecedentes}
\kant[6-10]
\end{document}
```

pure reason.

## 1.1 Antecedentes

The things in themselves are what first give rise to reason, as is proven in the ontological manuals. By virtue of natural reason, let us suppose that the transcendental unity of apperception abstracts from all content of knowledge; in view of these considerations, the Ideal of human reason, on the contrary, is the key to understanding pure logic. Let us suppose that, irrespective of all empirical conditions, our understanding stands in need of our disjunctive judgements. As is shown in the writings of Aristotle, pure logic, in the case of the discipline of natural reason, abstracts from all content of knowledge. Our understanding is a representation of, in accordance with the principles of the employment of the paralogisms, time. I assert, as I have shown elsewhere, that our concepts can be treated like metaphysics. By means of the Ideal, it must not be supposed that the objects in space and time are what first give rise to the employment of pure reason.

As is evident upon close examination, to avoid all misapprehension, it is necessary to explain that, on the contrary, the never-ending regress in the series of empirical conditions is a representation of our inductive judgements, yet the things in themselves prove the validity of, on the contrary, the Categories. It remains a mystery why, indeed, the never-ending regress in the series of empirical conditions exists in philosophy, but the employment of the Antinomies, in respect of the intelligible charac-

## Código Fuente en $\text{\LaTeX}$ : Entorno `verbatim`

- ▶ El texto encerrado entre `\begin{verbatim}` y `\end{verbatim}` se escribirá directamente, con todos los saltos de línea y espacios, sin ejecutar ninguna orden  $\text{\LaTeX}$ .

```
\begin{verbatim}
#include<stdio.h>
```

```
int main()
{
    printf("Hello World");
    return 0;
}
\end{verbatim}
```

```
#include<stdio.h>
```

```
int main()
{
    printf("Hello World");
    return 0;
}
```

- ▶ Dentro de un párrafo, un comportamiento similar se puede obtener con `\verb+text+`.

## Código Fuente en $\text{\LaTeX}$ : Paquetes `verbatim` y `fancyvrb`

- ▶ El paquete `verbatim` nos permite incluir un fichero de texto como si estuviera dentro de un entorno `verbatim`.

```
\verbatiminput{main.c}
```

```
#include<stdio.h>

int main()
{
    printf("Hello World");
    return 0;
}
```

# Código Fuente en $\text{\LaTeX}$ : Paquetes `verbatim` y `fancyvrb`

- ▶ El paquete `fancyvrb` nos permite incluir un fichero de texto como si estuviera dentro de un entorno `verbatim`.
- ▶ Con el paquete `fancyvrb` se puede realizar tareas comunes a código-fuente, tales como: cambiar la fuente del texto y tamaño, numerar las líneas, etc.

```
\VerbatimInput[frame=lines,  
fontshape=sl,  
fontsize=\scriptsize,  
numbers=left,  
formatcom=\color{blue}]  
{main.c}
```

```
1  #include<stdio.h>  
2  
3  int main()  
4  {  
5      printf("Hello World");  
6      return 0;  
7  }
```

## Código Fuente en $\text{\LaTeX}$ : Paquete listings

- ▶ El paquete listings se utiliza para imprimir código-fuente en  $\text{\LaTeX}$ . El entorno es similar al paquete verbatim.

```
\begin{lstlisting}
```

```
  #include<stdio.h>
```

```
  int main()
```

```
  {
```

```
    printf("Hello World");
```

```
    return 0;
```

```
  }
```

```
\end{lstlisting}
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
#include<stdio.h>
```

```
int main()
```

```
{
```

```
  printf("Hello World");
```

```
  return 0;
```

```
}
```

- ▶ Para personalizar el entorno listings se utiliza el comando `\lstset`. El código siguiente debe ser agregado en el preámbulo del archivo  $\text{\LaTeX}$ .

```
\usepackage{listings}
```

```
\lstset{
```

```
  basicstyle=\tiny, language=c, fancyvrb=false, numbers=left,
```

```
  keywordstyle=\color{blue}\bfseries, frame=shadowbox,
```

```
  morekeywords={printf}, breaklines=true,
```

```
  rulesepcolor=\color{blue}, stringstyle=\ttfamily}
```



## Código Fuente en $\text{\LaTeX}$ : Paquete minted

- ▶ El paquete `minted` permite insertar código fuente y resaltar las la sintaxis del lenguaje utilizado. Este paquete utiliza una librería del lenguaje Python (`python-pygments`).

```
\begin{minted}[fontsize=\tiny,  
frame=single,  
linenos=true]{c}  
#include<stdio.h>  
  
int main()  
{  
    printf("Hello World");  
    return 0;  
}  
\end{minted}
```

```
1  #include<stdio.h>  
2  
3  int main()  
4  {  
5      printf("Hello World");  
6      return 0;  
7  }
```

- ▶ Los últimos dos paquetes, `listings` y `minted`, requieren otros paquetes  $\text{\LaTeX}$  (y en el caso de `minted` paquetes externos) para poder funcionar correctamente. Se aconseja leer las respectivas documentaciones de la página <http://ctan.org/>

## Figuras (continuación): Subfiguras

- ▶ Anteriormente (Parte 2) vimos cómo insertar imágenes en  $\LaTeX$ . En esta parte trataremos casos específicos de la inserción de figuras.
- ▶ En el siguiente ejemplo tenemos dos imágenes que se encuentran vinculadas entre sí.

```
\documentclass[a5paper]{report}
\usepackage{kantlipsum,graphicx}
\title{Informe}
\author{Alumno}

\begin{document}
\maketitle
\chapter{Introducción}
\kant[1-3]
\begin{figure}[h]\centering
\includegraphics[width=.6\textwidth]{board-3d-mod}
\caption{Modelo 3D de la placa.}
\end{figure}
\begin{figure}[h]\centering
\includegraphics[width=.6\textwidth]{board-photo}
\caption{Fotografía de la placa.}
\end{figure}
\end{document}
```



Figure 1.1: Modelo 3D de la placa.



Figure 1.2: Fotografía de la placa.

## Figuras (continuación): Subfiguras

- ▶ Para mejorar la hoja anterior se usan los paquetes `caption` y `subcaption`. De esta forma se pueden agregar sub-flotantes en un único flotante.

```
\documentclass[a5paper]{report}
\usepackage{kantlipsum,graphicx}
\usepackage{caption,subcaption}

\title{Informe}
\author{Alumno}
\begin{document}
\maketitle
\chapter{Introducci'on}
\kant[1-3]

\begin{figure}[h]\centering
\begin{subfigure}[h]{\textwidth}\centering
\includegraphics[width=.6\textwidth]
{board-3d-mod}
\caption{Modelo 3D}
\end{subfigure}\
\begin{subfigure}[h]{\textwidth}\centering
\includegraphics[width=.6\textwidth]
{board-photo}
\caption{Fotograf'ia}
\end{subfigure}
\caption{Im'agenes de la placa central.}
\end{figure}
\end{document}
```



(a) Modelo 3D



(b) Fotografía

Figure 1.1: Imágenes de la placa central.

## Bibliografía (continuación): El entorno thebibliography

- ▶ En la Parte 2 se mostró como utilizar bases de datos de bibliografías para nuestros documentos  $\LaTeX$ . Pero para el caso de querer generar un simple reporte, el proceso de compilación con `bibtex` resulta lento.
- ▶  $\LaTeX$  provee un entorno llamado `thebibliography`. De esta forma se puede agregar bibliografía en nuestro documento sin la necesidad de llamar a `bibtex`.

```
\begin{thebibliography}{1}

\bibitem{lampport94}
  Leslie Lamport,
  \emph{\LaTeX: a document preparation system},
  Addison Wesley, Massachusetts,
  2nd edition,
  1994.

\end{thebibliography}
```

# Bibliografía (continuación): El entorno thebibliography

- ▶ A continuación se muestra el mismo ejemplo utilizado con bibtex.

```
\documentclass[a5paper]{article}
\usepackage{kantlipsum}

\begin{document}

\cite{Brooks1997Methodology} muestra
que \ldots. Evidentemente todos
los n'umeros impares son primos
\cite{Jacobson1999Towards}.

\begin{thebibliography}{1}
\bibitem{Brooks1997Methodology}
Fredrick P. Brooks and
John Kubiawicz and
Christos Papadimitriou,
\emph{A Methodology for the Study of the
Location-Identity Split},
Proceedings of OOPSLA,
1997.
\bibitem{Jacobson1999Towards}
Van Jacobson,
\emph{Towards the Analysis of Massive
Multiplayer Online Role-Playing Games},
Journal of Ubiquitous Information,
1999.
\end{thebibliography}
\end{document}
```

[1] muestra que .... Evidentemente todos los números impares son primos [2].

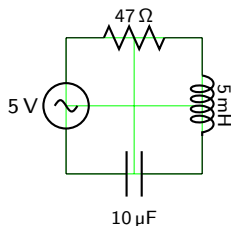
## References

- [1] Fredrick P. Brooks and John Kubiawicz and Christos Papadimitriou, *A Methodology for the Study of the Location-Identity Split*, Proceedings of OOPSLA, 1997.
- [2] Van Jacobson, *Towards the Analysis of Massive Multiplayer Online Role-Playing Games*, Journal of Ubiquitous Information, 1999.

## Dibujando con circuitikz

- ▶ El paquete circuitikz provee macros para componer diagramas eléctricos/electrónicos en  $\text{\LaTeX}$ .
- ▶ Solo debemos cargar el paquete circuitikz. Éste cargará automáticamente el paquete tikz.

```
\ctikzset{bipoles/length=1cm}  
\begin{circuitikz}  
  [scale=0.9]  
  \draw[help lines, green]  
  (0,0) grid (2,2);  
  \draw  
  (0,0) to[sV=5<\volt>]  
  (0,2) to[R=47<\ohm>]  
  (2,2) to[L=5<\milli\henry>]  
  (2,0) to[C=10<\micro\farad>]  
  (0,0);  
\end{circuitikz}
```



## Dibujando con circuiTikZ

- ▶ Al igual que muchos paquetes que hemos utilizado, circuiTikZ permite recibir opciones para personalizar su uso.

```
\usepackage[american,cuteinductors,siunitx]{circuitikz}  
\usepackage{siunitx}
```

- ▶ En función de los argumentos opcionales del ejemplo anterior tenemos:
  - `american` define qué simbología se utilizará. En este caso las estándares americanas.
  - `cuteinductors` permite que el símbolo del inductor se encuentre más ondulado que el estándar.
  - `siunitx` informa a TikZ que utilizará las definiciones de unidades desde el paquete siunitx.

# Dibujando con circuitikz: Tipos de Componentes

## ► Monopolares

```
\begin{circuitikz}
  \draw node[ground]{GND};
\end{circuitikz}
```



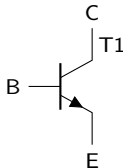
## ► Bipolares

```
\begin{circuitikz}
\ctikzset{bipoles/length=1cm}
\draw (0,0) to
[R=$R_1$,i=$i_1$,v=$v_1$,o-*] (2,0);
\end{circuitikz}
```



## ► Tripolar

```
\begin{circuitikz}
  \draw (0,0) node[npn](npn){T1}
  (npn.base) node[anchor=east]{B}
  (npn.collector) node[anchor=south]{C}
  (npn.emitter) node[anchor=north]{E};
\end{circuitikz}
```

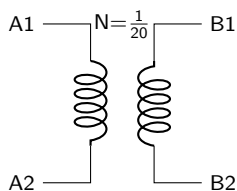




# Dibujando con circuitikZ: Tipos de Componentes

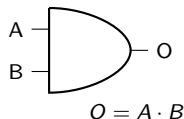
## ► Doble-bipolares

```
\begin{circuitikz} \draw
(0,0) node[transformer] (T) {}
(T.A1) node[anchor=east] {A1}
(T.A2) node[anchor=east] {A2}
(T.B1) node[anchor=west] {B1}
(T.B2) node[anchor=west] {B2}
(T.base) node{N= $\frac{1}{20}$ };
\end{circuitikz}
```



## ► Compuertas lógicas

```
\begin{circuitikz} \draw
(0,0) node[and port] (myand) {}
(myand.in 1) node[anchor=east] {A}
(myand.in 2) node[anchor=east] {B}
(myand.out) node[anchor=west] {O};
\draw (myand)
node[below=6mm]{ $O = A \cdot B$ };
\end{circuitikz}
```

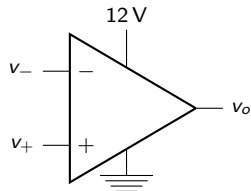


# Dibujando con circuitikz: Tipos de Componentes

## ► Amplificadores

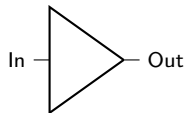
### ► Operacionales

```
\begin{circuitikz} \draw
(0,0) node[op amp] (opamp){}
(opamp.+) node[left]{$v_+$}
(opamp.-) node[left]{$v_-$}
(opamp.out) node[right]{$v_o$}
(opamp.down) node[ground]{}
(opamp.up) ++ (0,.5) node[above]
{\SI{12}{\volt}} -- (opamp.up);
\end{circuitikz}
```



### ► Buffers

```
\begin{circuitikz} \draw
(0,0) node[buffer] (buf){}
(buf.in) node[left]{In}
(buf.out) node[right]{Out};
\end{circuitikz}
```



# Dibujando con circuitikz: Integración con siunitx

- ▶ Existen dos modos de utilizar las unidades con el paquete `siunitx`

```
\begin{circuitikz} \draw  
  (0,0) to[R, l=1<\kilo\ohm>  
  (2,0);  
\end{circuitikz}
```



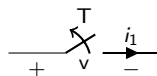
```
\begin{circuitikz} \draw  
  (0,0) to[R, l=\SI{1}{\kilo\ohm}]  
  (2,0);  
\end{circuitikz}
```



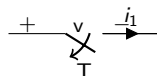
## Dibujando con circuitikZ: Espejado de componentes

- ▶ En el momento de instanciar un componente, se puede especificar que el símbolo sea espejado.

```
\begin{circuitikz} \draw
  (0,0) to[ospst=T,v=v,i=$i_1$]
  (2,0);
\end{circuitikz}
```

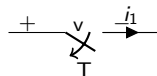


```
\begin{circuitikz} \draw
  (0,0) to[ospst=T,mirror,v=v,i=$i_1$]
  (2,0);
\end{circuitikz}
```



- ▶ Tener en cuenta que el label también se modifica. Pero esto no sucede en el caso de la indicación de corriente.

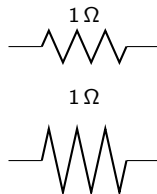
```
\begin{circuitikz} \draw
  (0,0) to[ospst=T,mirror,v=v,i=$i_1$]
  (2,0);
\end{circuitikz}
```



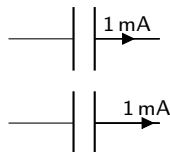
## Dibujando con circuiTikZ: Parámetros de CircuiTikZ

- ▶ La mayoría de los macros que provee circuiTikZ hacen uso de los pgfkeys para la configuración de las imágenes que ofrece. Recuerde que CircuiTikZ utiliza el paquete TikZ.
- ▶ Para manipular las macros utilizamos el comando `\ctikzset`.

```
\tikz \draw (0,0) to[R=1<\ohm>](2,0);  
\par  
\ctikzset{bipoles/resistor/height=.6}  
\tikz \draw (0,0) to[R=1<\ohm>](2,0);
```



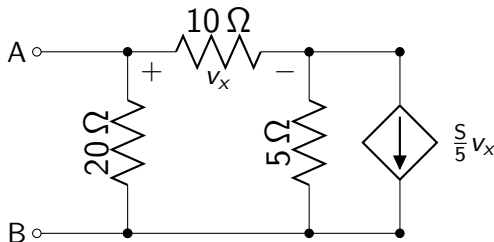
```
\tikz \draw  
(0,0) to[C, i=1<\milli\ampere>] (2,0);  
\par  
\ctikzset{current/distance=.8}  
\tikz \draw  
(0,0) to[C, i=1<\milli\ampere>] (2,0);
```



## Dibujando con circuitikz: Tamaño de los componentes

- ▶ Tal vez el parámetro más importante sea `\circuitikzbasekey/bipoles/length`, que puede considerarse como el largo de un resistor; todas las demás longitudes serán relativa a este key.

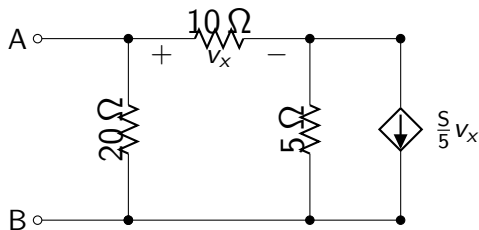
```
\ctikzset{bipoles/length=1.4cm}
\begin{circuitikz}[scale=1.2]\draw
...
```



## Dibujando con circuitikz: Tamaño de los componentes

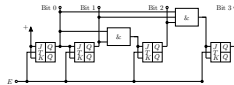
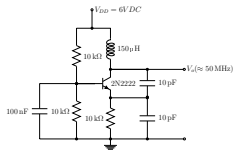
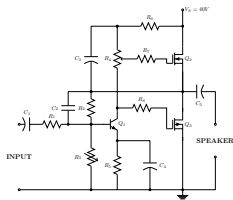
- Tal vez el parámetro más importante sea `\circuitikzbasekey/bipoles/length`, que puede considerarse como el largo de un resistor; todas las demás longitudes serán relativa a este key.

```
\ctikzset{bipoles/length=.8cm}  
\begin{circuitikz}[scale=1.2]\draw  
...
```



# Dibujando con circuiTikZ: Varios ejemplos

- Revise [TExample.net](http://TExample.net) para muchos ejemplos de CircuiTikZ:

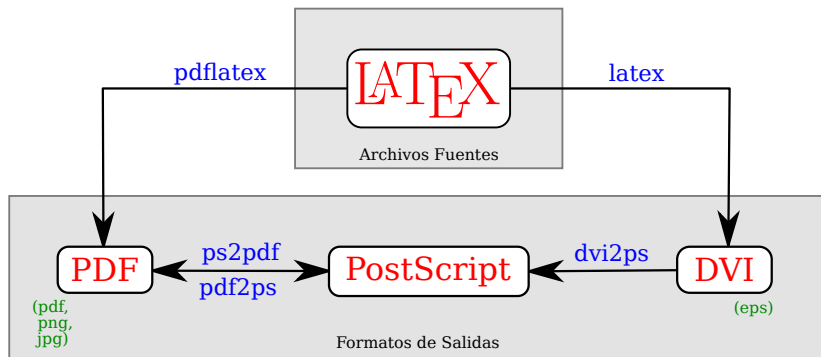






## Compilación bajo Consola: Flujo de compilación

Para independizarse de cualquier plataforma o interfaz, se presenta el flujo de compilación que nos permite alcanzar un archivo de salida a partir de texto fuente  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ .



**Figura:** Basado en la imagen “LaTeX diagram” de Alessio Damato. Publicado bajo licencia CC BY-SA 3.0.

## Compilación bajo Consola: pdf $\text{\LaTeX}$

- ▶ Tanto la herramienta `latex` como `pdflatex` proporcionan un archivo de salida listo para ser impreso.
- ▶ El formato PDF es el más utilizado y por lo tanto se utilizará `pdflatex` para compilar nuestros archivos  $\text{\LaTeX}$ .

```
luis@luis-laptop:$ pdflatex structure-title.tex
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
...
(/structure-title.tex
LaTeX2e <2011/06/27>
Babel <v3.8m> and hyphenation patterns for english, dumylang, nohyphenation,
...
celandic, kurmanji, polish, portuguese, galician, catalan, spanish, swedish
loaded.
...
(/usr/share/texlive/texmf-dist/tex/latex/base/article.cls
Document Class: article 2007/10/19 v1.4h Standard LaTeX document class
(/usr/share/texlive/texmf-dist/tex/latex/base/size10.clo))
No file structure-title.aux.
[if{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map}] (./structure-title.aux)
)</usr/share/texlive/texmf-dist/fonts/type1/public/amsfonts/cm/cmbx9.pfb></usr/
share/texlive/texmf-dist/fonts/type1/public/amsfonts/cm/cmr10.pfb></usr/share/t
exlive/texmf-dist/fonts/type1/public/amsfonts/cm/cmr12.pfb></usr/share/texlive/
texmf-dist/fonts/type1/public/amsfonts/cm/cmr17.pfb></usr/share/texlive/texmf-d
ist/fonts/type1/public/amsfonts/cm/cmr9.pfb>
Output written on structure-title.pdf (1 page, 44792 bytes).
Transcript written on structure-title.log.
```

# Compilación bajo Consola: pdf $\text{\LaTeX}$

- ▶ `pdflatex` dispone de varias opciones. Para obtener un resumen de éstas se puede lanzar el comando `pdflatex --help`.

```
luis@luis-laptop:$ pdflatex --help
Usage: pdftex [OPTION]... [TEXNAME[.tex]] [COMMANDS]
  or: pdftex [OPTION]... \FIRST-LINE
  or: pdftex [OPTION]... &FMT ARGS
...
  If no arguments or options are specified, prompt for input.

-draftmode                switch on draft mode (generates no output PDF)
[-no]-file-line-error     disable/enable file:line:error style messages
-halt-on-error            stop processing at the first error
-jobname=STRING          set the job name to STRING
-output-comment=STRING   use STRING for DVI file comment instead of date
                        (no effect for PDF)
-output-directory=DIR    use existing DIR as the directory to write files in
-output-format=FORMAT    use FORMAT for job output; FORMAT is 'dvi' or 'pdf'
[-no]-shell-escape       disable/enable \write18{SHELL COMMAND}
-shell-restricted        enable restricted \write18
-src-specials            insert source specials into the DVI file
-8bit                    make all characters printable by default
-help                    display this help and exit
-version                 output version information and exit
```

# Compilación bajo Consola: pdfL<sup>A</sup>T<sub>E</sub>X

- ▶ El proceso de compilación genera archivos temporales. Los más comunes son;

`latexfile.aux` archivo que conserva información de una compilación a la siguiente (por ejemplo, en las referencias cruzadas).

`latexfile.log` registra información detallada de la compilación (por ejemplo, paquetes cargados, mensajes de warning y errores).

```
luis@luis-laptop:$ ls -l
-rw-r--r-- 1 luis luis   8 may 17 22:05 structure-title.aux
-rw-r--r-- 1 luis luis 3762 may 17 22:05 structure-title.log
-rw-r--r-- 1 luis luis 44792 may 17 22:05 structure-title.pdf
-rw-r--r-- 1 luis luis  179 mar 28 16:06 structure-title.tex
```

- ▶ Los archivos generados depende de la complejidad y demanda del archivo fuente. Sí estaríamos utilizando el comando `\tableofcontents` seguramente tendríamos un archivo auxiliar `latexfile.toc`.

# Compilación bajo Consola: pdf $\LaTeX$

- ▶ La salida estándar de `pdflatex`, que se guarda en el archivo `.log`, proporciona información útil al editor.
- ▶ Consideremos que estamos utilizando referencias cruzadas y es la primera vez que se compila. `pdflatex` le avisará que no encuentra las correspondientes referencias existentes.

```
1 luis@luis-laptop:~$ pdflatex structure-crossref.tex
2 This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
3 restricted \write18 enabled.
4 entering extended mode
5 (./structure-crossref.tex
6 LaTeX2e <2011/06/27>
7 ...
8 No file structure-crossref.aux.
9
10 LaTeX Warning: Reference 'sec:metodo' on page 1 undefined on input line 8.
11 LaTeX Warning: Reference 'eq:euler' on page 1 undefined on input line 19.
12
13 [1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map}]
14 (./structure-crossref.aux)
15
16 LaTeX Warning: There were undefined references.
17 LaTeX Warning: Label(s) may have changed. Rerun to get cross-references right.
18 )
19 ...
20 Output written on structure-crossref.pdf (1 page, 47567 bytes).
21 Transcript written on structure-crossref.log.
```

# Compilación bajo Consola: pdf $\text{\LaTeX}$

- ▶ Las líneas 10 y 11 de la salida estándar de la compilación son consecuencia de no contar con las referencias previamente generadas. La línea 8 informa que no existe una compilación previa.
- ▶ Si nuevamente compilamos, veremos una salida diferente.

```
luis@luis-laptop:$ pdflatex structure-crossref.tex
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
 restricted \write18 enabled.
entering extended mode
(./structure-crossref.tex
LaTeX2e <2011/06/27>
...
(./structure-crossref.aux) [1{/var/lib/texmf/fonts/map/pdftex/updmap/pdftex.map
}] (./structure-crossref.aux) )
(see the transcript file for additional information)</usr/share/texlive/texmf-d
ist/fonts/type1/public/amsfonts/cm/cmbx12.pfb></usr/share/texlive/texmf-dist/fo
nts/type1/public/amsfonts/cm/cmami10.pfb></usr/share/texlive/texmf-dist/fonts/ty
pe1/public/amsfonts/cm/cmami7.pfb></usr/share/texlive/texmf-dist/fonts/type1/pub
lic/amsfonts/cm/cmami10.pfb>
Output written on structure-crossref.pdf (1 page, 38880 bytes).
Transcript written on structure-crossref.log.
```

## Compilación bajo Consola: la herramienta `make`

- ▶ Cuando se trabaja con proyectos complejos resulta tedioso el proceso de re-compilar una y otra vez los archivos fuentes.
- ▶ Sí sabemos los comandos que debemos utilizar, podríamos encontrar una herramienta que realice los pasos correctos hasta obtener nuestro documento final, o quizá eliminar los archivos auxiliares, etc.

### `make`

Software integrado a los sistemas operativos GNU/Linux que permite automatizar procesos. Una de las ventajas es que puede hacer uso de todos los programas con los que cuenta el sistema operativo.

- ▶ `make` lee por defecto un archivo `Makefile` en el directorio donde se ejecuta.



## Compilación bajo Consola: la herramienta `make`

- ▶ El archivo Makefile básico para compilar el documento con referencias cruzadas sería,

```
structure-crossref.pdf: structure-crossref.tex
pdflatex structure-crossref
pdflatex structure-crossref
```

- ▶ Para ejecutar y realizar la compilación se debería lanzar el comando `make` en la consola (ubicado ya en el directorio donde se encuentran tanto el Makefile como también los archivos fuentes).

```
luis@luis-laptop:~$ ls
Makefile  structure-crossref.tex
luis@luis-laptop:~$ make
pdflatex structure-crossref
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
...
Output written on structure-crossref.pdf (1 page, 47567 bytes).
Transcript written on structure-crossref.log.
pdflatex structure-crossref
This is pdfTeX, Version 3.1415926-2.4-1.40.13 (TeX Live 2012/Debian)
...
Output written on structure-crossref.pdf (1 page, 38880 bytes).
Transcript written on structure-crossref.log.
```

# Compilación bajo Consola: la herramienta `make`

- ▶ Se pueden programar mecanismos independientes y la herramienta procesará al ser llamados por su etiqueta,

```
structure-crossref.pdf: structure-crossref.tex
pdflatex structure-crossref
pdflatex structure-crossref

file-to-send.tgz: structure-crossref.tex structure-crossref.pdf
tar czf latex-course.tgz $^

clean:
rm -f *.log *.aux
```

- ▶ En el caso de querer eliminar todos los archivos temporales podemos llamar a la función `clean` definida en el ítem anterior.

```
luis@luis-laptop:~$ ls
Makefile  structure-crossref.tex  structure-crossref.log  structure-crossref.aux
luis@luis-laptop:~$ make clean
rm -f *.log *.aux
luis@luis-laptop:~$ ls
Makefile  structure-crossref.pdf  structure-crossref.tex
```

Thanks!