Capítulo 1

Cosas que debe saber

1.1. LaTeX

LaTeX es un programa de computador creado por Donald E. Knuth. Sirve para componer texto y fórmulas matemáticas. Knuth empezó a escribir el motor de composición LaTeX en 1977 para investigar el potencial de los equipos de impresión digital que estaban empezando a usarse en la industria de la impresión en aquel tiempo; en concreto tenía la esperanza de poder revertir la tendencia de calidad tipográfica en declive que él vio afectar a sus propios libros y artículos.

LaTeX se pronuncia "látej" o "latech". La "j" surge del alfabeto griego donde X es la letra "j" o "ji". LaTeX es también la primera sílaba de la palabra griega $\tau\epsilon\xi\eta\nu$ (arte).

1.2. Lo básico

1.2.1. Autor, maquetador y compositor

Para publicar algo, los autores dan su manuscrito mecanografiado a una editorial. Uno de sus maquetadores decide el aspecto del documento (anchura de columna, tipografías, espacio ante y tras cabeceras, etc.). El maquetador escribe sus instrucciones en el manuscrito y luego se lo da al compositor o cajista, quien compone el libro siguiendo esas instrucciones.

Un maquetador humano suele interpretar qué pretendía el autor mientras escribía el manuscrito. Decide sobre las cabeceras de los capítulos, las citas, los ejemplos, las fórmulas, etc., basándose en su conocimiento profesional y en el contenido del manuscrito.

En un entorno LaTeX, LaTeX represnta el papel del maquetador. Pero LaTeXes "sólo" un programa, y por tanto necesita más supervisión. El autor tiene que proporcionar información adicional para describir la estructura lógica de su trabajo. Tal información se escribe entre el texto como "órdenes LaTeX".

Esto es bastante diferente del enfoque visual o WYSIWYG[1] que sigue la mayoría de los procesadores de texto modernos, como *Open Office Writer* o *Microsoft Word*. Con estos programas, los autores especifican el aspecto del documento interactivamente mientras escriben texto en el computador. Así pueden ver en la pantalla cómo aparecerá el trabajo final cuando se imprima.

Cuando use LaTeX no suele se posible ver el aspecto final del texto mientras lo escribe, pero tal aspecto puede verse en pantalla tras procesar el archivo mediante LaTeX. Entonces pueden hacerse correcciones antes de enviar el documento a la impresora para tener una copia en papel.

1.2.2. Maquetación

La maquetación (diseño tipográfico) es un arte. Los autores sin habilidad a menudo cometen errores de formato al suponer que maquetar es mayormente una cuestión de estética, "Si un documento luce bien artísticamente, está bien diseñado". Pero como un documento se escribe para ser leído y no colgado en una galería de arte, su legibilidad es mucho más importante que su aspecto. Ejemplos.

- El tamaño de los tipos y la numeración de las cabeceras debe escogerse para que la estructura de capítulos y secciones quede clara al lector.
- La longitud de línea debe ser suficientemente corta para no cansar a los ojos del lector, pero suficientemente larga para llenar la página apropiadamente.

Con sistemas WYSIWYG, los autores a menudo generan documentos agradables estéticamente pero con muy poca o muy inconsistente estructura. LaTeX impide tales errores de formato forzando al autor a declarar la estructura *lógica* del documento. LaTeX escoge entonces la composición más adecuada.

1.2.3. Ventajas y desventajas

Cuando gente del mundo WYSIWYG se encuentra con usuarios de LaTeX, a menudo discuten "las ventajas de LaTeX sobre un procesador de textos normal" o lo contrario. Lo mejor que puede hacer cuando un debate tal comienza es mantenerse al margen, pues tales discusiones a menudo se salen de control. Pero a veces uno no puede escapar.

Pues he aquí algunos argumentos. Las principales ventajas de LaTeX sobre procesadores de texto normales son las siguientes:

- Se dispone de composiciones diseñadas profesionalmente, lo que hace que un documento parezca realmente "impreso".
- El soporte para la composición para la composición de fórmulas matemáticas es muy adecuado.

[1] What you see is what you get: Lo que ve es lo que consigue

- Los usuarios sólo tienen que apredner unas pocos órdenes fáciles de entender, que especifican la estructura lógica del documento. Casi nunca necesitan preocuparse del aspecto real del documento.
- Es fácil generar incluso estructuras complejas, como notas al pie, referencias, índces o bibliografías.
- Existen paquetes libres (incluso gratuitos) que facilitan muchas tareas tipográficas especializadas, no soportadas directamente por el LaTeX básico. Por ejemplo, hay disponibles paquetes para incluir gráficos o para componer bibliografías según normas precisas.
- LaTeX incita a los autores a escribir textos bien estructurados, porque así trabaja LaTeX.
- LaTeX es libre y muy portable. Por tanto, puede ejecutarse en casi cualquier plataforma informática disponible.

LaTeX tiene también algunas desventajas, y supongo que me es un poco encontrar alguna notable, aunque estoy seguro de que otros le podrán hablar de cientos

- Aunque pueden ajustarse algunos parámetros dentro de una cierta composición del documento, el diseño de una nueva composición completa es difícil y lleva mucho tiempo.
- Puede que su aprendiz nunca llegue a entender, a pesar de ciertos primeros pasos prometedores, a comprender el concepto de marcado lógico.

1.3. Estructura del archivo de entrada

La entrada para LaTeXes un archivo de texto puro. Puede crearlo con cualquier editor de texto. Contiene el texto del documento, así como las órdenes que dirán a LaTeXcómo componer el texto.

Cuando LaTeX procesa un archivo de entrada, espera que siga una cierta estructura. Así, todo archivo de entrada ha de comenzar con la orden

```
\documentclass{...}
```

Esto indica qué tipo de documento pretende escribir. Después, puede incluir órdenes que influyen el estilo de todo el documento, o puede cargar paquetes que añaden nuevas prestaciones al sistema LaTeX. Para cargar un paquete use la orden

```
\usepackage{...}
```

Cuando todo el trabajo de preparación está hecho, comience a escribir el cuerpo del texto con la orden

```
\begin{document}
```

El área entre `\documentclass` y `\begin{document}` se llama *preámbulo*.

Ahora escriba el texto mezclado con órdenes LaTeX útiles. Al final del documento añada la orden

```
\end{document}
```

que dice a LaTeX que termine el trabajo. Cualquier cosa que siga a esta orden será ignorada.

La figura 1.1 muestra el contenido de un archivo LaTeX.

```
\documentclass{article}
\begin{document}
Mi primer documento en \LaTeX.
\end{document}
```

Figura 1.1: Un archivo LaTeXmínimo.

1.3.1. Una típica seisón de consola

Como se decía antes, LaTeX por sí mismo viene sin GUI (interfaz gráfica de usuario) ni botones para pulsar. Es un programa de procesamiento que "mastica" y "digiere" su archivo de entrada para "generar" su(s) archivo(s) de salida. Algunas instalaciones de LaTeX ofrecen una interfaz gráfica donde usted puede escribir y compilar su archivo de entrada (TeXnicCenter, TexMaker, Kile). En otros sistemas puede requerirse la escritura de ciertas órdenes, de modo que he aquí cómo lograr que LaTeXcompile su archivo de entrada en un sistema basado en texto.

1. Edite/cree su archivo de entrda LaTeX. Este archivo debe ser texto puro. Puede crearlo con cualquier editor de texto, pero asegúrese de que guarda el archivo con formato *Texto Puro*. Al escoger un nombre para el archivo, póngale como extensión `.tex`

2. Ejecute LaTeX en su archivo de entrada. Si tiene éxito aparecerá un archivo `.dvi`. Puede que necesite ejecutar LaTeX varias veces para que los índices y todas las referencias internas queden correctamente definidas. Si su archivo de entrada tiene un error, LaTeX se lo dirá y parará el procesamiento de su archivo de entrada. Escriba `CTRL + D` para volver a la lína de órdenes.

   ```
   latex miDocumento.tex
   ```

3. Finalmente, la orden `PDFLaTeX` le permite compilar el archivo directamente en pdf.

   ```
   pdflatex miDocumento
   ```

1.3.2. Espacio

- LaTeX trata los caracteres "en blanco", tales como el espacio en blanco o el tabulador, uniformemente como "espacio".
- Varios caracteres consecutivos en blanco se tratan como uno solo.
- Una línea vacía entre dos líneas de texto define el fin de un párrafo.
- Varias líneas vacías se tratan igual que *una sola* línea vacía.

A la izquierda está el texto del archivo de entrada, y a la derecha está la salida formateada.

Ejemplo 1.1.

```
No importa si usted deja
uno o varios          espacios
tras una palabra.


Una línea vacía comienza
un nuevo párrafo.
```

No importa si usted deja uno o varios espacios tras una palabra.
Un línea vacía comienza un nuevo párrafo.

1.3.3. Caracteres especiales

Los siguientes símbolos son caracteres reservados que tienen un significado especial bajo LaTeXo no están disponibles en todas las tipografías. Si los pone directamente en su texto, normalmente no se imprimirán, sino que obligarán a LaTeX a hacer cosas que no se pretendían.

Observación 1 (Caracteres reservados)**.**

$ % ^ & _ { } ~ \

Como verá, se pueden usar estos caracteres en sus documentos añadiendo una diagonal invertida como prefijo:

Ejemplo 1.2.

```
\# \$ \% \^{} \& \_
\{ \} \~{} $\backslash$
```

 # $ % ^ & _
*{ } ~ *

Los demás símbolos y muchos más pueden imprimirse con órdenes especiales en fórmulas matemáticas, o como acentos.

Observación 2 (Contradiagonal)**.**

La contra diagonal (diagonal invertida) no puede introducirse añadiendo otra contradiagonal delante (\\); esta orden se usa para saltar de línea[a].

[a]Se debe usar la orden `\backslash` en su lugar.

1.3.4. Comandos en LaTeX

Las órdenes LaTeX son sensibles a mayúsculas, y adoptan uno de los dos formatos siguientes:

- Comienzan con una contradiagonal y luego tienen un nombre que consiste sólo en letras. Los nombres de orden terminan con un espacio, un número o cualquier otra "no-letra".
- Consisten en una contradiagonal y exactamente una no-letra.

Observación 3.

LaTeX prescinde del espacio en blanco tras las órdenes. Si quiere conseguir un espacio tras una orden, tiene que poner { }.

Ejemplo 1.3.

```
He leído que Knuth divide a la
gente que trabaja con \TeX{} en
\TeX{}nicos y \TeX pertos.\\
Hoy es \today.
```

He leído que Knuth divide a la gente que trabaja con TeX en TeXnicos y TeXpertos.
Hoy es 30 de junio de 2015.

Algunas órdenes requieren un argumento, que tiene que ponerse entre llaves { } tras el nombre de la orden. Algunas órdenes soportan argumentos opcionales, que se añaden tras el nombre de la orden entre corchetes [ ].

Ejemplo 1.4. *Los siguientes ejemplos usan algunas órdenes LaTeX, no se preocupe por ellos; se explicarán más adelante:*

```
Puede \textit{fiarse} de mí.

Por favor, comienza una nueva
línea justo aquí\newline Gracias
```

Puede fiarse *de mí.*
Por favor, comienza una nueva línea justo aquí
Gracias

1.3.5. Comentarios

Cuando LaTeX encuentra un caracter % al procesar un archivo de entrada, presciende del resto de la línea actual, el salto de línea y todo el espacio en blanco al comienzo de la línea siguiente.

Esto puede usarse para escribir notas en el archivo de entrada, que no se mostraán en la versión impresa.

Ejemplo 1.5.

```
Este es un % estúpido
% Mejor: instructivo <----
ejemplo: Supercal%
ifragilíst%
icoespialidoso
```

Este es un ejemplo: Supercalifragilísticoespialidoso

El caracter % también puede usarse para dividir líneas largas en la entrada donde no se permiten espacios ni saltos de línea.

Para comentarios más largos puede usar el entorno `comment` proporcionado por los paquetes `comment` o `verbatim`. Esto significa que tiene que añadir la línea `\usepackage{verbatim}`, o la línea `\usepackage{comment}` al preámbulo del documento, como se explica en el ejemplo siguiente:

Ejemplo 1.6.

```
Esto es otro
\begin{comment}
bastante estúpido,
pero útil
\end{comment}
ejemplo para insertar
comentarios en su texto.
```

Esto es otro ejemplo para insertar comentarios en su texto.

Observación 4.

Tenga en cuenta que eso no funciona dentro de entornos complejos, como por ejemplo los matemáticos.

1.4. El aspecto del documento

1.4.1. Clase de documento

La primera información que LaTeX necesita saber cuando procesa un archivo de entrada es el tipo de documento que el autor quiere crear. Esto se indica con la orden `\documentclass`

```
\documentclass[opciones]{clase}
```

Aquí *clase* indica el tipo de documento por crear. La tabla 1.1 lista las clases de documentos explicadas en esta introducción. El parámetro *opciones* personaliza el comportamiento de la clase. Las opciones tienen que separarse por comas. Las opciones más comunes para las clases de documento habituales se listan en la tabla 1.2

article	para artículos en revistas científicas, informes breves, documentación de programas, invitaciones, etc.
proc	para actas, basado en la clase article.
minimal	es lo más pequeña posible. Solamente establece un tamaño de página y una fuente (tipo de letra). Se usa principalmente para depurar errores.
report	para informes más largos que contienen varios capítulos, pequeños libros, tesis, etc.
book	para libros reales.
beamer	para diapositivias. La clase usa letras grandes.

Cuadro 1.1: Clases de documento.

Ejemplo 1.7. *Un archivo de entrada para un documento LaTeX podría empezar con la línea*

```
\documentclass[11pt, twoside, a4paper]{article}
```

que manda a LaTeX componer el documento como un artículo *con un tamaño de fuente básica de* once puntos*, y producir un documento adecuado para imprimir a* doble cara *en* papel A4.

. . . . . .

1.4.2. Paquetes

Mientras escribe su documento, probablemente halle que hay algunas áreas donde el LaTeX básico no puede resolver su problema. Si quiere incluir gráficos, texto en color o código fuente de un archivo en su docuemnto, necesita mejorar las capacidades de LaTeX. Tales mejoras se introducen con *paquetes*. Los paquetes se activan con la orden

```
\usepacakge[opciones]{paquete}
```

donde *paquete* es el nombre del paquete y *opciones* es una lista de palabras clave que activan funciones especiales del paquete.

1.4.3. Proyectos grandes

Cuando trabaje en proyectos grandes, puede servirle dividir el archivo de entrada en varias partes que puede reunir al compilarlo. La forma más práctica de hacerlo es:

```
\intpu{nombre-de-archivo}
```

`10pt, 11pt, 12pt`	Establece el tamaño de la principal fuente del documento. Si no se especifica ninguna opcionón, se aplica 10pt.
`a4paper,letterpaper`	Define el tamaño del papel. el tamaño por default es letterpaper.
`fleqn`	Dispone las fórmulas destacadas hacia la izquierda en vez de centradas.
`leqno`	Coloca los números de las fórmulas a la izquierda en vez de a la derecha.
`titlepage, notitlepage`	Indica si tras el título del documento debe empezarse una página nueva o no. La clase `article` no comienza página nueva por omisión, mientras que `report` y `book` sí la tienen.
`onecolumn, twocolumn`	Dice a LaTeX que componga el documento de una, o dos, columnas respectivamente.
`openright`	Hace que los capítulos comiencen en páginas derechas. Esto no funciona con la clase `article`, pues no entiende de capítulos.

Cuadro 1.2: Opciones de clases de documento.