

Guía rápida de KOMA - Script

Joaquín Ataz López

Versión 1.0. Marzo de 2015

Índice

1. Visión global de KOMA- Script	2
1.1. ¿Qué es KOMA- Script?	2
1.2. ¿Qué podemos hacer con KOMA- Script? o ¿En qué modifica KOMA- Script a los tipos de documento estándar?	3
1.3. Compatibilidad entre KOMA- Script y los tipos de documento estándar	4
1.4. Sobre la presente guía	4
2. Funcionamiento de las opciones de KOMA- Script	5
2.1. Formato general de las opciones de KOMA- Script	5
2.2. Atribución de valor a las opciones después de cargado el estilo o paquete	7
3. Control del diseño básico de la página	8
3.1. Qué significa en la presente guía «diseño básico de la página»	8
3.2. Algoritmos para calcular el tamaño y ubicación del bloque de texto de una página	9
3.2.1. Cálculo de las proporciones de la página mediante el método de la división	9
3.2.2. Cálculo del diseño mediante el método del círculo	11
3.3. Opciones y comandos de KOMA- Script que afectan al diseño de la página	11
3.3.1. Las opciones básicas: BCOR y DIV	11
3.3.2. Opciones adicionales que afectan al funcionamiento de DIV	13
3.3.3. Indicación directa del tamaño del área de texto, del encabezado o del pie de página	14
3.3.4. Otras opciones relacionadas con el diseño de la página	15
3.4. Selección del tamaño de la fuente base	17
3.5. Los elementos preliminares del documento	18
3.5.1. Título y abstract	18
3.5.2. Índice general del documento	20
3.6. El cuerpo del documento	22
3.6.1. Espaciado vertical entre párrafos	22
3.6.2. Notas al pie	23
3.6.3. Partes, capítulos, secciones	25
3.6.4. Estilos de página, encabezados y pies	29

3.6.5. Tablas y figuras flotantes	32
3.7. Modificación del formato aplicado por defecto a los distintos elementos del documento	32
4. Otros comandos y entornos que no existen en \LaTeX estándar	32
4.1. Citas o dichos	32
4.2. Descripciones alternativas a «description»: El entorno labeling	33
4.3. Modificar los márgenes de un fragmento de texto: El entorno addmargin	33
4.4. Otros comandos	34

1. Visión global de KOMA- Script

1.1. ¿Qué es KOMA- Script?

KOMA- Script consiste en una serie de paquetes y estilos de \LaTeX que, en conjunto constituyen una *revisión completa* del funcionamiento de los tipos estándar de documento. El resultado de todo ello es que:

- Los documentos generados con KOMA- Script tienen un aspecto visual distinto al de los documentos “clásicos” de \LaTeX . Casi diría que el aspecto es más «actual».
- El autor dispone de un control sobre los distintos aspectos formales de su documento mucho mayor del que otorgan los tipos estándar de documento de \LaTeX .

He hablado —y volveré a hacerlo a lo largo de este texto— de los tipos estándar de documento de \LaTeX . Ello es porque KOMA- Script propone cuatro tipos de documento “nuevos”, en sustitución de los cuatro tipos de documento estándar de \LaTeX ¹:

\LaTeX estándar	KOMA- Script	Pensado para
book	scrbook	Libros
article	scrartcl	Artículos científicos
report	scrreprt	Informes
letter	scrletter	Cartas

Pero, tal y como dije al principio, KOMA- Script es un conjunto de paquetes y estilos. Lo que significa que aunque KOMA- Script está pensado para que se usen sus tipos de documentos, hay ciertos aspectos que, en cuanto dependen de paquetes específicos, pueden usarse en los documentos de \LaTeX estándar, si se carga expresamente el correspondiente paquete. Y en particular, una de las más potentes utilidades de KOMA- Script, el diseño de la página, descansa en un paquete —«typearea»— que se puede utilizar con los estilos estándar de \LaTeX .

¹ En realidad \LaTeX estándar ofrece dos tipos adicionales de documento: «proc» y «slides». Pero tales tipos de documento —aparte de que no son tan usados como el resto— no pueden considerarse “documentos de propósito general”.

1.2. ¿Qué podemos hacer con KOMA- Script? o ¿En qué modifica KOMA- Script a los tipos de documento estándar?

A lo largo del presente documento se explicarán con cierto detalle algunas de las utilidades de KOMA- Script. No todas, pues ello requeriría un documento excesivamente largo. Pero en este primer capítulo indicaré en grandes líneas las que a mi juicio son las principales aportaciones de KOMA- Script, de manera que el lector sepa si le merece la pena o no seguir leyendo.

Con KOMA- Script podemos:

- Cambiar en cualquier punto del documento cualquiera de sus características principales que en los documentos estándar son inmutables tales como, por ejemplo, el tamaño del papel o el tamaño de la letra base.
- Controlar y modificar el diseño general de las páginas del documento; esto es: el tamaño y ubicación, dentro de la hoja de papel, del bloque de texto.
- Controlar el formato de numerosos elementos individuales del documento que \LaTeX estándar no permite modificar tales como, por ejemplo, los encabezados, los pies de página, los títulos de las unidades estructurales, etc.
- Seleccionar para nuestro documento cualquier tamaño de papel y cualquier tamaño de fuente base, y cambiarlos, si se desea, en cualquier punto.
- Un conjunto de comandos y entornos adicionales a los de los documentos estándar que se pueden considerar de propósito general, pues se basan en un análisis de tareas o elementos que suele haber en los textos y para los que \LaTeX estándar no tiene un comando o entorno específico.
- En fin: una serie de pequeñas y sutiles (o no tan sutiles) modificaciones en prácticamente todos los elementos de los documentos (título, índice, notas a pie, unidades estructurales...), que cabe catalogar en diferentes grupos:
 - (a) Algunas —la mayoría— pretenden permitir al autor del documento controlar el resultado final; permitiéndole configurar aspectos que en los documentos estándar no son configurables.
 - (b) Otras se limitan a automatizar tareas que habitualmente se realizan y para las que no existe un comando u opción.
 - (c) Un tercer grupo, proporciona comandos o entornos nuevos que extienden la funcionalidad del sistema.
 - (d) Un último grupo, en fin, se basa en consideraciones de los autores de KOMA- Script relativas a lo que es tipográficamente más correcto o adecuado. Estas últimas, modificaciones, como es lógico, van en gustos y habrá quien prefiera la solución de los tipos de documento estándar. Aunque hay que decir que el alto nivel de control que KOMA- Script concede al autor, lleva a que en estos casos casi siempre sea posible volver a la solución de \LaTeX estándar.

La suma de todas estas características hace que, como dije al principio, en los documentos creados con KOMA- Script el autor disponga de un mayor control sobre el documento y el resultado tenga “otro aire”, que antes he calificado de “más actual” o “moderno”; aunque esto, claro está, no deja de ser una percepción personal.

Para los autores europeos KOMA- Script ofrece además una ventaja menor, y es que, al haberse desarrollado esta herramienta en Alemania, en general los valores por defecto de las distintas opciones de KOMA- Script se ajustan más a los estándares europeos. Y así, por ejemplo, el tamaño de papel por defecto no es el formato «letter» de L^AT_EX estándar, que en Europa apenas se utiliza, sino el formato DIN A4, bastante más extendido por estos lares.

1.3. Compatibilidad entre KOMA- Script y los tipos de documento estándar

KOMA- Script ha sido diseñado para lograr la mayor compatibilidad con los tipos de documento estándar. Todas las opciones que dichos documentos admiten son conocidas por KOMA- Script, así como todos los comandos, entornos y unidades estructurales.

La idea es que usar KOMA- Script resulte tan sencillo como sea posible. De hecho si en un documento ya escrito, basado en alguno de los estilos estándar de L^AT_EX simplemente cambiamos la indicación del tipo de documento por el tipo equivalente en KOMA- Script (por ejemplo, cambiamos «book» por «scrbook») y recompilamos, lo normal es que no se produzca ningún error. Y si tras ello examinamos el resultado comprobaremos que todo está en su lugar y que todos los comandos y entornos han surtido su efecto. Lo único que habrá cambiado son aquellos aspectos en los que los valores por defecto de KOMA- Script difieren de los de los documentos estándar como, por ejemplo, la fuente, el tamaño del papel, el tamaño de la letra base, la distribución del texto en la hoja, la posición de los encabezados o pies de página, etc. E incluso estos aspectos sólo cambiarán si en el documento que originalmente habíamos basado en uno de los estilos clásicos, no se había indicado específicamente nada respecto de ellos.

Por ejemplo: los estilos clásicos usan por defecto una fuente de 10 puntos de tamaño, mientras que KOMA- Script, por defecto, aplica un tamaño de 11 puntos. Pero si en nuestro documento original se había indicado algún tamaño de fuente mediante la opción correspondiente, éste no cambiará al aplicar un estilo propio de KOMA- Script; incluso aunque el tamaño de fuente expresamente especificado fuera exactamente el tamaño por defecto.

1.4. Sobre la presente guía

KOMA- Script es muy amplio. Su ayuda oficial, en inglés, consta de 419 páginas (la versión de 14-01-2015). La ayuda es, por otra parte, bastante completa, y supongo que cualquier usuario de L^AT_EX lo suficientemente avanzado está en condiciones de acceder a ella y localizar la información que le interese, incluso aunque su nivel de inglés o de alemán sea relativamente bajo.

En la ayuda oficial, sin embargo, yo echaba en falta un apartado general sobre para qué usar KOMA- Script. De ahí que me haya decidido a escribir el presente texto que, en el fondo, he escrito para mí mismo. En él:

- Expongo con cierto desarrollo los aspectos de KOMA- Script que me parecen más importantes. En particular el control del diseño de página y la gestión de fuentes.
- Expongo de una forma mucho más sucinta, alguna de las utilidades que KOMA- Script ofrece para aspectos más concretos del documento. En esta exposición me entretengo más en aquellas cuestiones que a mí me parecen más interesantes, lo cual —claro está— tiene que ver con el tipo de documentos que yo mismo desarrollo.

Hay utilidades de KOMA-Script que en este documento ni siquiera se mencionan. A veces porque no he terminado de entenderlas tal y como las explica la ayuda (piénsese que mi nivel de inglés no es muy allá y mis conocimientos tipográficos tampoco). Otras porque la explicación ocuparía demasiado espacio, y esta guía no pretende ofrecer información detallada sino más bien exponer en conjunto las posibilidades de KOMA-Script, de tal manera que quien las vea sugerentes pueda sumergirse en la herramienta indagando en la ayuda de la misma.

Para poder seguir esta guía hay que tener en cuenta que en ella se presupone que el lector conoce \LaTeX y se maneja bien con los documentos estándar, de modo que, cuando se habla de un elemento de \LaTeX no se explica en qué consiste, y aunque a veces, para que quede más claro lo que KOMA-Script aporta, se empieza explicando cómo son las cosas en los documentos estándar, eso no es, desde luego, lo más normal. La mayor parte de las veces se asume que el lector sabe cómo funcionan las cosas sin KOMA-Script, y simplemente se informa de qué utilidad adicional se incorpora, que “problema” se pretende resolver o qué aspecto tipográfico resulta modificado.

En el momento de escribir esta guía, la última versión de KOMA-Script es la 3.17. Pero como esa versión es muy reciente, y aún no se ha incorporado a los repositorios de las diferentes distribuciones de linux, de tal modo que para usarla hay que instalarla a mano, he preferido basar el presente documento en la versión 3.12, que es la que se incluye en las distribuciones actuales de TeXlive.

2. Funcionamiento de las opciones de KOMA-Script

Gran parte de las utilidades de KOMA-Script se indican como opciones del tipo de documento. Conviene por lo tanto, en primer lugar, exponer algunas peculiaridades de las opciones que los documentos escritos con alguno de los estilos incorporados por KOMA-Script esperan recibir.

2.1. Formato general de las opciones de KOMA-Script

Las opciones básicas de los tipos de documento estándar de \LaTeX consisten en una palabra o secuencia de texto que tiene un significado especial para el compilador y a la que por ello podemos denominar “palabra simbólica” en el sentido de que simboliza algo. Así, por ejemplo, las opciones «10pt», «11pt» o «12pt» para representar el tamaño base de la letra del documento, o «a4paper» para representar el tamaño del papel. En ocasiones, incluso, para representar una utilidad que puede estar activada o no, como por ejemplo, la impresión a doble cara, se utilizan dos opciones distintas: una para indicar la activación (en nuestro ejemplo «twoside») y otra para indicar la no activación («oneside»).

El problema que tiene el usar este tipo de opciones, asociada cada una de ellas a un único valor, es el de que si para cada posibilidad necesitamos una palabra distinta, un estilo que fuera altamente configurable, necesitaría cientos, o miles de palabras simbólicas para representar todas sus posibilidades... lo que en la práctica haría inviable su aprendizaje y, por tanto, su uso. Por ello este sistema de indicación de opciones sólo es adecuado cuando las opciones a indicar son pocas, como ocurre en los tipos estándar de documento de \LaTeX , que son muy

poco configurables y, por lo tanto, necesitan pocas opciones.

Pero KOMA-Script tiene entre sus finalidades la de dar al autor de documentos un mayor control sobre los mismos y, por lo tanto, ese formato de indicación de opciones no es adecuado. Por ello los documentos escritos con alguno de los estilos incorporados por KOMA-Script utilizan opciones con el siguiente formato:

`Nombre=Valor`

El valor, por otra parte, puede ser de muy distintos tipos. A veces es un valor numérico, a veces es una longitud, y otras veces es una palabra simbólica, o sea: una palabra que KOMA-Script reconoce como asociada a cierto valor o a cierta acción. La palabra, en estos casos, hay que introducirla tal cual; pues debe recordarse que L^AT_EX diferencia entre mayúsculas y minúsculas. Aunque en este punto debe tenerse en cuenta que en muchos casos KOMA-Script asocia dos o más palabras a un mismo valor, lo que, en teoría, hace más flexible el manejo de las opciones.

Por ejemplo: para indicar que en el índice general debe aparecer una entrada para la bibliografía, la opción denominada «toc» admite los valores «bibliography» y «bib», y para indicar que, además, esa entrada debe numerarse como si se tratara de una sección más, se admiten hasta cuatro palabras distintas: «numberedbibliography», «bibliographynumbered», «numberedbib» y «bibnumbered».

Como se puede ver estas cuatro palabras son las distintas combinaciones que se pueden obtener juntando la palabra «numbered», con las dos palabras que representan la entrada en el índice para la bibliografía. Este procedimiento de formar nombres para las opciones, para sus valores, o para los comandos, es bastante habitual en KOMA-Script: produce denominaciones muy largas, pero bastante claras sobre su significado.

Un caso concreto de opciones que reciben como valor una palabra simbólica son las que sirven para activar o desactivar cierta utilidad. Gran parte de las opciones de KOMA-Script son de este tipo y pueden recibir los valores que se indican en la tabla 1.

Opciones
Si/No

Palabra	Efecto
true	Activa la opción
on	Activa la opción
yes	Activa la opción
false	Desactiva la opción
off	Desactiva la opción
no	Desactiva la opción

Tabla 1: Valores posibles para las opciones tipo Si/No

De otro lado, para cada opción hay dos tipos de valores por defecto:

1. Valor por defecto para el caso de que la opción no se especifique.
2. Valor por defecto para el caso de que se especifique la opción, pero no se le atribuya ningún valor concreto.

Por ejemplo: la opción «twocolumn», que controla si el texto del documento se distribuye en una sola columna o en dos columnas, por defecto no está activada, es decir: si no se indica expresamente el texto constará de una sola columna. Pero si se indica la opción, y no se le

asigna ningún valor, se asumirá que se le quería asignar el valor «true», y la utilidad de doble columna quedará activada.

Todas las opciones tienen algún valor por defecto. Pero sólo en algunas es posible especificar el nombre de la opción sin asignarles ningún valor concreto. Eso ocurre, por ejemplo, en la mayor parte de las opciones del tipo «Si/No». Pero hay otras opciones que si se establecen y no se les atribuye un valor concreto, provocan que se genere un error durante la compilación del documento.

Estimo que conocer de cada opción el tipo de valor que espera, cuál es su valor por defecto, y qué ocurre si se especifica la opción pero no se le asigna ningún concreto valor, es algo relevante. Por ello, cada vez que se presente una opción, en este documento se mostrará un cuadro como el siguiente en el que se recoge la información básica sobre la opción:

Opción: Nombre de la opción
<ul style="list-style-type: none">▪ <i>Tipo de valor</i>: Número, Longitud, Palabra simbólica o Si/No▪ <i>Valor por defecto</i>: Valor cuando no se establece la opción▪ <i>Si no se le asigna un valor</i>: Valor que se le asigna cuando se establece la opción y no se le atribuye un valor concreto, o indicación de que en tal caso se produciría un error

2.2. Atribución de valor a las opciones después de cargado el estilo o paquete

En un paquete o estilo normal de L^AT_EX las opciones sólo se pueden indicar en el momento de carga del mismo, lo que tiene el inconveniente de que ello impide cambiar el valor de dicha opción una vez que el estilo o el paquete se han cargado; lo cual es especialmente grave en estilos, ya que el estilo de un documento se carga en su primera línea, cuando todavía no se conocen otros valores, que se cargan más tarde, y que tal vez debieran interactuar con dicha opción e influir en su resultado.

Por ello muchos paquetes ofrecen algunos comandos que permiten cambiar el valor de las opciones incluso después de que el paquete o el estilo se haya cargado.

KOMA-Script también lo hace, y en concreto ofrece para ello dos comandos que permiten alterar el valor de cualquier opción en cualquier punto del documento. Se trata de los comandos:

<pre>\KOMAOPTIONS{lista de opciones} \KOMAoption{opción}{lista de valores}</pre>
--

El primero de los comandos permite modificar (o establecer por primera vez) una opción indicando el nombre de la opción, el signo igual y su valor. Para indicar varias opciones, éstas se separan unas de otras mediante una coma. Por ejemplo para establecer el papel al tamaño DIN A4 y la fuente a 12 puntos habría que escribir:

<pre>\KOMAOPTIONS{paper=A4,fontsize=12pt}</pre>

El segundo comando está pensado para ciertas opciones (poco corrientes) que admiten simultáneamente más de un valor. El primer argumento es el nombre de la opción y el segundo la

lista de valores, separados unos de otros por comas. Pero nada impide que se use este segundo comando para establecer el valor de una sola opción. Por ejemplo:

```
\KOMAOption{fontsize}{12pt}
```

En ambos comandos hay que tener en cuenta que:

1. Si se introduce un nombre de opción desconocido, o un tipo de valor inadecuado para la opción, se generará un error durante la compilación.
2. Si entre los valores de una opción debe figurar el signo igual (=) o una coma (,), dicho valor debe encerrarse entre llaves.

Más adelante se comprenderá mejor la tremenda flexibilidad que aporta esta posibilidad de indicar el valor de las opciones del estilo en un momento posterior al de la carga del mismo. Ya de momento podemos percatarnos de que con ella se hace posible algo que en las clases estándar de \LaTeX no lo es: generar un documento en el que alguno de los aspectos controlados por las opciones del tipo de documento (tamaño del papel, tamaño de la letra base, impresión a doble cara...) pueda cambiar en un punto dado del documento, de tal manera que, por ejemplo, cabe la posibilidad de que las tres primeras páginas sean de tamaño A4 y la cuarta de tamaño A5.

Cuando la opción cuyo valor se cambia afecta globalmente a las páginas como, por ejemplo, el tamaño del papel, o la impresión a doble cara, un cambio en dicho valor implica que se insertará, en el punto en el que se establece el cambio, un salto de página.

3. Control del diseño básico de la página

3.1. Qué significa en la presente guía «diseño básico de la página»

Por *diseño básico de la página* entiendo la fijación del tamaño del bloque de texto de la página: su altura y su anchura; así como la ubicación de dicho bloque en la hoja.

El diseño básico de la página, por lo tanto, afecta al establecimiento de sus márgenes. Y de hecho, en un programa procesador de texto del tipo WYSIWYG, este diseño básico se determina simplemente fijando, en primer lugar, el tamaño de la hoja, y, a continuación, las medidas de los cuatro márgenes: superior, inferior, izquierdo y derecho. Pero hacer las cosas así, aunque parece muy sencillo, significa renunciar a la excelencia tipográfica. Porque después de varios siglos de imprimir libros las diferentes tradiciones tipográficas llegaron a la conclusión de que los márgenes de una página no deben fijarse de modo arbitrario. La correcta legibilidad de la página exige que dentro de ella existan ciertas proporciones, y que para fijar el tamaño de los márgenes, además de respetar dichas proporciones, se tomen en consideración muchos factores tales como el tamaño y diseño de la fuente, el tamaño del interlineado, la existencia o no de cabeceras o pies de página y lo recargados que estos estén, etc.

De ahí que en los documentos estándar de \LaTeX no se le pregunte al autor qué márgenes desea, sino que estos se fijan automáticamente; aunque se pueden alterar manipulando directamente las medidas internas de las páginas, tales como « \textwidth » o « \textheight », lo cual

implica por su parte que también en \LaTeX se pueden cometer aberraciones en el diseño de las páginas. Pero a diferencia de los programas WYSIWYG, que obligan al autor a decidir sobre aspectos de los que posiblemente no entienda, las aberraciones en \LaTeX hay que cometerlas a caso hecho.

KOMA-Script ofrece a los autores un mecanismo que permite influir en el diseño de la página, pero garantizando que el resultado respetará las proporciones óptimas de las páginas. Este mecanismo está incorporado en el paquete `«typearea»` que es cargado automáticamente por los estilos de KOMA-Script, pero que puede ser también usado por los estilos clásicos de \LaTeX . En cuyo caso las opciones que a continuación se explican serían, no opciones del estilo, sino opciones del paquete. También estaría disponible la posibilidad de modificar el valor de las opciones después de cargado el paquete mediante los comando en la sección 2.2

3.2. Algoritmos para calcular el tamaño y ubicación del bloque de texto de una página

Para comprender las opciones que permiten controlar el diseño básico de la página, ayuda bastante conocer, aunque sea sólo por encima, los procedimientos que utiliza KOMA-Script para determinar las proporciones óptimas de la página. Los cuales son el llamado *método de la división* (procedimiento por defecto) y el que la documentación oficial de KOMA-Script denomina *método del círculo* o también *diseño clásico*. Por defecto se usa el método de la división.

3.2.1. Cálculo de las proporciones de la página mediante el método de la división

Este método parte de una serie de asunciones, basadas en determinados estudios de diseño y composición de páginas, según los cuales lo ideal es que el margen inferior y el superior guarden una proporción de 2 a 1, y el margen exterior y el interior (en documentos impresos a doble cara) guarden la misma proporción.

Partiendo de esa asunción de “proporciones idóneas”, KOMA-Script realiza los siguientes pasos para establecer el tamaño de los márgenes:

1. En primer lugar, del ancho total de la página, sustrae la cantidad de espacio que se perderá tras la encuadernación. Esa longitud, a la que en inglés se denomina *binding correction*, se resta del lado izquierdo, en documentos impresos a una sola cara, y, en documentos impresos a doble cara, del lado interior que es, en las páginas impares, el lado izquierdo, y en las pares, el derecho.

La cantidad de espacio que hay que reservar para la encuadernación no puede ser conocida de antemano, pues depende del tipo de encuadernación. Por ello se le debe indicar expresamente a KOMA-Script, cosa que se hace con la opción BCOR (véase más adelante).

2. El ancho resultante se divide en un número de columnas al que, de momento, llamaremos DIV. En la figura 1 se muestra un ejemplo en el que se ha usado para DIV el valor de 9.
3. A continuación se divide la página verticalmente en el mismo número DIV de filas.

Con ello se crea una especie de cuadrícula que permite ubicar con comodidad la zona de texto respetando las proporciones que se consideran óptimas para los márgenes, tal y como se

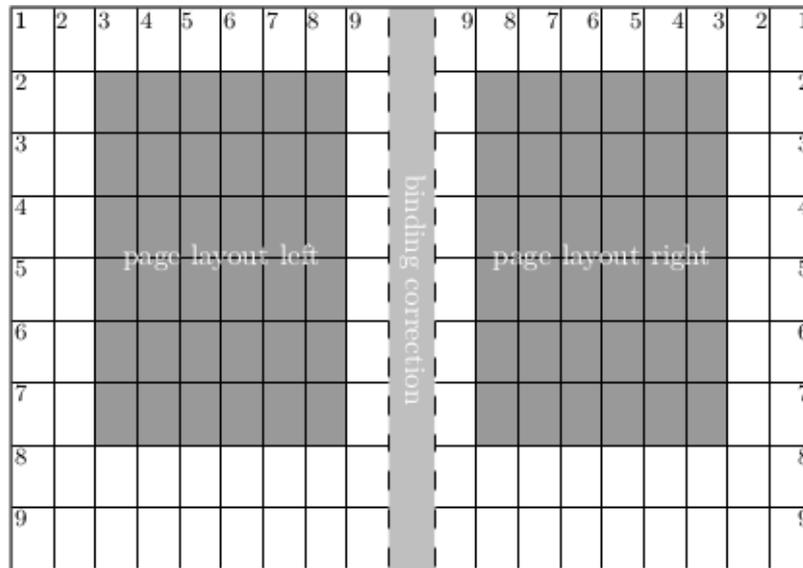


Figura 1: Método de la división para calcular los márgenes óptimos

puede ver en la figura 1. En ella se ha reservado la primera fila como margen superior y las dos últimas filas como margen inferior, el cual es, por lo tanto, el doble, lo que respeta la proporción que antes se indicó que KOMA- Script asume como óptima. Lo mismo ocurre con los márgenes exterior e interior.

Esta forma de calcular los márgenes implica que éstos serán mayores o menores dependiendo del número de filas y columnas que formen la cuadrícula. Aumentando el valor de DIV habrá más filas y columnas, pero cada una de ellas será más estrecha, por lo que también los márgenes serán más reducidos. Por el contrario, si reducimos el valor de DIV aumentará el tamaño de los márgenes y disminuirá el del bloque de texto.

Hay pues una relación directa entre el tamaño de DIV y el del bloque de texto, una relación inversa entre el tamaño de DIV y el tamaño de los márgenes. De manera que para conseguir un diseño óptimo nos basta con calcular cuál es el valor idóneo de DIV.

KOMA- Script decide cuál es ese valor idóneo de dos maneras distintas, según en qué caso se encuentre:

1. Si el papel es de tamaño DIN A4 y la fuente utilizada es de 10, 11 o 12 puntos, se asigna un valor de DIV predeterminado que será de 8, 10 o 12, según cuál sea el tamaño de la fuente.
2. En los demás casos se calcula un valor para DIV en el que se toman en consideración todos los factores que KOMA- Script considera relevantes.

Es importante señalar que el segundo método suele ofrecer mejores resultados que el primero, pues los factores relevantes para determinar el tamaño óptimo de la zona de texto no son sólo el tamaño del papel y el de la fuente, sino que influyen también, por ejemplo, el diseño de la fuente (las hay más anchas y más estrechas), el interlineado o el tamaño del encabezado y del pie de página. Sin embargo el primer procedimiento —que a veces ofrece resultados peores—

es el que se aplica a la mayoría de los documentos pues, al menos en Europa, el tamaño normal del papel es DIN A4 y la fuente suele oscilar precisamente entre 10 y 12. Aunque la cuestión no es preocupante, pues podemos forzar a KOMA- Script a que utilice el segundo procedimiento incluso aunque se den las condiciones para el primero.

3.2.2. Cálculo del diseño mediante el método del círculo

Este procedimiento está bastante menos explicado que el anterior en la documentación de KOMA- Script y sólo se utiliza cuando expresamente se le pide. Se basa, según parece, en algunos diseños medievales de libros y por ello se le llama también “*método clásico*”. Según la documentación de KOMA- Script está especialmente recomendado cuando la altura de la zona de texto es igual a la anchura de la página.

3.3. Opciones y comandos de KOMA- Script que afectan al diseño de la página

3.3.1. Las opciones básicas: BCOR y DIV

Determinación del espacio reservado para la encuadernación

Opción: BCOR

- *Tipo de valor:* Longitud
- *Valor por defecto:* 0
- *Si no se le asigna un valor:* Se produce un error

En páginas que se unirán físicamente a otras para formar un libro o folleto, es conveniente sustraer de la anchura real de la hoja la porción que será usada para la encuadernación o el grapado y que. Para indicar la longitud de esa zona se utiliza la opción BCOR:

Control del tamaño de la zona de texto

Opción: DIV

- *Tipo de valor:* Número entero o palabra simbólica
- *Valor por defecto:* Depende de varios factores
- *Si no se le asigna un valor:* La palabra simbólica «`calc`»

DIV es la opción de la que principalmente depende el tamaño relativo de los márgenes y de la zona de texto de la página. Puede consistir en un número entero o en una palabra simbólica. Cuando consiste en un número entero dicho número representan el valor por el que se dividirá el ancho y el largo de la página para determinar la anchura de los márgenes. El cómo se hace esto exactamente ha sido explicado en la sección 3.2.1. Para que surta efecto, el valor de DIV debe ser superior a 4. Cuanto mayor sea el valor de DIV más grande será la zona de la página reservada al texto y más reducidos los márgenes. Mi consejo al lector es que vaya experimentando con distintos valores de DIV para apreciar la diferencia. En mi opinión el rango de valores de DIV

que proporcionan un documento con un diseño de página razonable oscila, si el tamaño de fuente es de 11 puntos, entre 10 y 18. Por debajo de 10 los márgenes son demasiado amplios para mi gusto y por encima de 18 las páginas quedan demasiado recargadas de letras.

El valor por defecto de DIV lo fija KOMA-Script atendiendo a si el tamaño del papel es o no DIN A4 y si el tamaño de la fuente es de 10, 11 o 12 puntos. Cuando se dan ambas circunstancias se asigna a DIV un valor predeterminado que depende del tamaño de la fuente: 8, para fuentes de 10 puntos, 10 para fuentes de 11 puntos y 12 para fuentes de 12 puntos. En los demás casos KOMA-Script calcula un valor óptimo para DIV atendiendo a una multitud de factores.

Además de indicar directamente el valor de DIV, es posible asignarle alguno de los siguientes valores simbólicos:

calc En mi opinión el valor más útil para DIV. Indicando «`DIV=calc`» se provoca que KOMA-Script recalculé el valor óptimo para DIV a la vista de todos los factores que tienen influencia en él: tamaño de papel, tipo de letra, diseño de la letra, interlineado, número de columnas, etc. Tras determinarse así el valor óptimo para DIV, inmediatamente se procede a recalcular el diseño de la página aplicando dicho valor.

Dado que para determinar cuál puede ser el valor óptimo de DIV hay que tomar en cuenta muchos factores, y dado también que en los estilos incorporados por KOMA-Script DIV es calculado en la primera línea del documento, cuando se carga el tipo de documento, es posible que el valor inicialmente calculado no sea el correcto, porque en las siguientes líneas del documento se altere alguno de los factores relevantes, como por ejemplo si se carga una fuente distinta a la fuente por defecto, o se altera el interlineado. Por ello, muchas veces conviene indicar «`\KOMAOPTIONS{DIV=calc}`» al final del preámbulo del documento; cuando ya se conocen todas las especificaciones que KOMA-Script toma en consideración para determinar cuál es el valor óptimo para DIV.

En documentos en los que KOMA-Script no “calcula” ningún valor para DIV, sino que éste es atribuido directamente dependiendo del tamaño de la letra, «`DIV=calc`» fuerza a que se busque el valor óptimo para DIV tomando en consideración no sólo el tamaño del papel y de la letra, sino el resto de factores que son relevantes.

classic Este valor provoca que se recalculé el tamaño del bloque de texto, pero usando el llamado método del círculo o método clásico. Véase la sección [3.2.2](#)

current Provoca que se recalculé el tamaño del bloque de texto, pero sin alterar el actual valor de DIV.

default Determina que se recalculé el tamaño del bloque de texto, asignando para DIV los valores predeterminados según el tamaño de papel y la fuente. Si para la combinación de papel y fuente no existe un valor predeterminado, «`default`» funcionará igual que «`calc`».

last Recalcula el tamaño y ubicación del bloque de texto usando el mismo valor de DIV que se usó en el último cálculo.

areaset Equivale a ejecutar el comando «`\areaset`» (que se explicará en la sección XXX), usando como valores para la anchura y altura, los valores actuales de «`\textwidth`» y «`\textheight`».

Cuando se carga por primera vez el paquete «`typearea`» se establece un valor inicial para DIV y para

BCOR que será el que se le haya indicado como opción, o el valor por defecto. Una vez establecido este valor, cualquier alteración de DIV provoca una llamada al comando interno de KOMA-Script «\typearea», cuyo formato es:

```
\typearea[BCOR]{DIV}
```

Aunque este comando ha sido diseñado para el uso interno del propio KOMA-Script, es posible usarlo directamente en el cuerpo del documento. E incluso está recomendado en un caso concreto: cuando se pretende cambiar simultáneamente los valores de BCOR y de DIV.

Para recalculer el bloque de texto con los valores actuales de BCOR y de DIV, el comando adecuado sería «\typearea[current]{last}». Pero como esta es una tarea que se realiza con cierta frecuencia, existe una versión abreviada que no requiere argumentos: «\recalctypearea».

3.3.2. Opciones adicionales que afectan al funcionamiento de DIV

Hasta ahora, al hablar del diseño de la página, hemos distinguido en ella exclusivamente tres zonas: el bloque de texto, los márgenes visibles, y el espacio destinado a la encuadernación.

Existen no obstante algunas partes de la página en las que podría dudarse si pertenecen a los márgenes o si pertenecen al bloque de texto. Decidir una cosa u otra afecta, como es lógico, al diseño de la página. Estas zonas son el encabezado, el pie de página y el espacio en el que se imprimen las notas al margen.

Las siguientes tres opciones nos permiten determinar si estas zonas se consideran o no parte del bloque de texto:

Opción: headinclude

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Opción: footinclude

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Opción: mpinclude

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Aunque el valor de estas opciones afecta al diseño óptimo de la página, el cambio en las mismas no provoca por sí mismo que se recalculen el tamaño del bloque de texto. Esto sólo

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

ocurrirá cuando se llame a los comandos «\typearea» o «\recalctypearea», o cuando se altere el valor de DIV.

3.3.3. Indicación directa del tamaño del área de texto, del encabezado o del pie de página

Indicación directa del tamaño del bloque de texto. El comando «\areaset» nos permite indicar directamente el tamaño del bloque de texto. Su formato es:

```
\areaset [BCOR] {Anchura}{Altura}
```

En los tres argumentos hay que indicar la unidad en la que se mide la longitud, al modo habitual de L^AT_EX. El bloque de texto así calculado se ubicará en la página teniendo en cuenta el valor de las opciones «BCOR», «headinclude», «footinclude», «mpinclude» y «twoside».

En general el uso de este comando es contrario a la filosofía que inspira a L^AT_EX, por lo que sólo está recomendado en casos especiales, como cuando una editorial regentada por personas tipográficamente incultas nos indica que los documentos deben tener cierto número de líneas por página y cierto número de caracteres por línea. En sistemas L^AT_EX, hacer esto es casi imposible, pero tal tipo de indicaciones se pueden interpretar como “no más de x caracteres o líneas”, o más probablemente “una media de x caracteres o líneas”. En este caso podríamos medir una página de ejemplo y usar «\areaset» para asegurarnos de que el resto de las páginas tengan las mismas dimensiones.

Calcular el número exacto de caracteres por línea sólo es posible cuando se usa una fuente de anchura fija. Y calcular el número de líneas exactas por página, exige un documento en el que no haya espaciado vertical extra entre párrafos, ni títulos, notas al pie, control de líneas viudas o huérfanas... nada más que párrafos normales. Pocos documentos se ajustan a tales especificaciones. Ni siquiera los documentos generados con procesadores de texto.

Ahora bien: si se interpreta que tales indicaciones en el sentido de que se refieren a una “media” de caracteres por línea y de líneas por página, es posible, en KOMA-Script diseñar la página que cumpla tales especificaciones. En la documentación de KOMA-Script se incluye el siguiente ejemplo, para obtener páginas de 60 caracteres de anchura y 30 líneas de texto:

```
% Cálculo de la anchura
\newlength{\CharsLX}
\settowidth{\CharsLX}{\texttt{1234567890}}
\setlength{\CharsLX}{6\CharsLX}

% Cálculo de la altura
\newlength{\LineasXXX}
\setlength{\LineasXXX}{\topskip}
\addtolength{\LineasXXX}{29\baselineskip}

% Fijación del bloque de texto
\areaset{\CharsLX}{\LineasXXX}
```

Indicación del tamaño del encabezado y pie de página. Podemos indicar estos tamaños bien como una longitud, bien como un número de líneas. Las opciones posibles por lo tanto son:

Opción: headlines

- *Tipo de valor:* Número (no necesariamente entero)
- *Valor por defecto:* 1.25
- *Si no se le asigna un valor:*

Opción: headheight

- *Tipo de valor:* Longitud
- *Valor por defecto:* Depende del interlineado
- *Si no se le asigna un valor:*

Opción: footlines

- *Tipo de valor:* Número (no necesariamente entero)
- *Valor por defecto:* 1.25
- *Si no se le asigna un valor:*

Opción: footheight

- *Tipo de valor:* Longitud
- *Valor por defecto:* Depende del interlineado
- *Si no se le asigna un valor:*

Obsérvese que en las cuatro opciones he dejado en blanco el apartado relativo a qué ocurre si la opción se establece, pero no se le asigna ningún valor concreto. Ello es porque, según mis pruebas, no se genera ningún error, pero no he encontrado en la documentación información sobre qué valor se asigna en estos casos. El valor no es exactamente el valor por defecto (para cuando no se establezca la opción) pues en mis pruebas se observa un cambio entre compilar sin mencionar la opción, y compilar mencionando la opción, pero sin aplicarle ningún valor.

3.3.4. Otras opciones relacionadas con el diseño de la página

Selección del tamaño del papel

Opción: paper

- *Tipo de valor:* Palabra simbólica o «*Altura:Anchura*»
- *Valor por defecto:* DIN A4
- *Si no se le asigna un valor:* DIN A4

Para tamaños no estándar de papel, podemos indicar las medidas separando la altura de la anchura mediante el carácter de los dos puntos. Por ejemplo `«paper=15cm:9cm»`. Pero tratándose de formatos conocidos de papel, podemos indicar directamente el nombre, en mayúsculas o en minúsculas. Formatos conocidos se consideran los formatos norteamericanos `«letter»`, `«legal»` o `«executive»` así como el nombre de cualquiera de los formatos ISO de las series A, B, C y D, como, por ejemplo, A3, A4, B5, etc.

La opción `«paper»` también puede asumir los valores `«landscape»` o `«seascape»`. Estos valores se usan para indicar el modo apaisado. Estas opciones se pueden usar directamente si el documento usa papel del tamaño DIN A4. Pero si se usa un tamaño de papel distinto, hay que utilizar dos veces la opción `«paper»`: la primera vez para indicar el tamaño de papel, y la segunda vez para indicar que debe usarse en el modo apaisado.

La diferencia entre usar `«landscape»` o `«seascape»` está en que en el primer caso `dvips` rota el papel -90 grados, y en el segundo la rotación es de +90 grados. A simple vista no se nota la diferencia, pero ésta es observable si se usa la opción `«pagesize»`.

Junto a la opción `«paper»` existe una opción muy parecida, denominada `«pagesize»`. Las características de esta opción son:

Opción: `pagesize`

- *Tipo de valor:* Palabra simbólica
- *Valor por defecto:* auto
- *Si no se le asigna un valor:* auto

Admito que no terminé de comprender la utilidad de esta opción y por lo tanto difícilmente la podré explicar. Sólo diré que los valores admisibles para esta opción son `«auto»`, `«automedia»`, `«false»`, `«no»`, `«off»`, `«dvipdfmx»`, `«dvips»` y `«pdftex»`.

Impresión a dos caras o a una sola cara. Por compatibilidad con los documentos estándar, existen dos opciones que controlan esta utilidad, aunque en realidad bastaría con una sola:

Opción: `twoside`

- *Tipo de valor:* Si/No* o `«semi»`
- *Valor por defecto:* Depende del tipo de documento
- *Si no se le asigna un valor:* true

Opción: `oneside`

- *Tipo de valor:* Si/No* o `«semi»`
- *Valor por defecto:* Depende del tipo de documento
- *Si no se le asigna un valor:* true

Estas opciones admiten los valores propios de una opción de tipo Si/No, así como el valor `«semi»` que provoca que la impresión sea a doble cara, y por lo tanto se generen páginas en

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

blanco, si es necesario, para garantizar que ciertas unidades estructurales empiezan en una página impar, pero sin embargo los márgenes de las páginas se calcularán como si la impresión fuera a una sola cara.

El valor por defecto de estas opciones depende del tipo de documento de que se trate: En documentos «scrbook» y «scrreprt» está activada por defecto la impresión a doble cara. En documentos tipo «scrartcl» y «sclttr2» por defecto la impresión es a una sola cara.

Impresión a doble columna

Opción: twocolumn

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Téngase en cuenta que esta opción se refiere al *diseño de página*, y no al *formato de columnas*. Ello quiere decir que en una misma página no pueden coincidir dos valores distintos para esta opción. O lo que es lo mismo: si en un punto determinado del documento cambiamos el valor de esta opción, se generará un salto de página. Por ello si lo que queremos es que *en la misma página* haya zonas a doble columna y otras a simple columna, en lugar de manipular los valores de esta opción, debermos usar algún paquete —como «multicol»— que ofrezca esa utilidad. Estos paquetes, además posiblemente permitan configurar aspectos que KOMA-Script (ni los documentos estándar de L^AT_EX) no permiten como, por ejemplo, el espaciado entre las columnas.

3.4. Selección del tamaño de la fuente base

El tamaño básico de la fuente es uno de los aspectos del documento que más influencia tiene para la determinación del tamaño óptimo del bloque de texto. Lo he dejado, no obstante, para el final de este capítulo debido a que mientras todos los aspectos mencionados hasta ahora venían establecidos por el paquete «typearea» que se puede cargar en documentos que no sigan alguno de los estilos de KOMA-Script, lo relativo al tamaño de la fuente no depende del paquete «typearea», y por lo tanto sólo es aplicable si se está usando alguno de los estilos incorporados por KOMA-Script.

En los documentos estándar de L^AT_EX sólo se admiten tres tamaños base para la fuente: 10, 11 o 12 puntos. El tamaño se indica, además, como opción del tipo de documento, y una vez establecido no se puede cambiar. Todo esto cambia en KOMA-Script donde esta utilidad está controlada por la opción «fontsize»:

1. Se puede usar cualquier tamaño de fuente como tamaño base.
2. El valor de «fontsize» —al igual que el del resto de opciones de KOMA-Script— se puede cambiar en cualquier punto del documento.

El formato de esta opción es el siguiente:

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

Opción: fontsize

- *Tipo de valor:* Longitud
- *Valor por defecto:* 11pt
- *Si no se le asigna un valor:* 11pt

Como en todas las longitudes de \LaTeX , hay que indicar la unidad de medida. Pero si no se indica ninguna se asume que el valor se ha indicado en puntos.

[Qué cambia KOMA-Script en los distintos elementos del documento]Cambios, mejoras y opciones adicionales que introduce KOMA-Script en los distintos elementos de un documento típico de \LaTeX

Este capítulo se refiere exclusivamente a los documentos generados con los estilos «`scrbook`», «`scrartcl`» y «`scrreprt`». El estilo «`scr1ttr2`» tiene tantas peculiaridades que es mejor examinarlo aparte.

3.5. Los elementos preliminares del documento

3.5.1. Título y abstract

Página del título y cubiertas preliminares. Si se dedica o no una página completa para el título, está controlado por la opción `titlepage`:

Opción: titlepage

- *Tipo de valor:* Si/No* o «`firstiscover`»
- *Valor por defecto:* Depende del tipo de documento
- *Si no se le asigna un valor:* `true`

El valor por defecto de esta opción depende del tipo de documento. En documentos «`scrbook`» y «`scrreprt`» se establece como «`true`» y en el caso de «`scrartcl`» se establece como «`false`».

Para KOMA-Script una página de título es algo distinto de una página de cubierta. La primera forma parte del documento y por lo tanto se ajusta al diseño general de páginas del mismo. La segunda tiene como misión exclusivamente proteger a la verdadera primera página de la suciedad; no forma parte del documento y no tiene por qué ajustarse a los valores generales del mismo, de modo que es posible redefinir sus márgenes redefiniendo (mediante «`\renewcommand`») los siguientes comandos:

```
\coverpagetopmargin
\coverpagebottommargin
\coverpageleftmargin
\coverpagerightmargin
```

Si queremos que nuestra página de título tenga la consideración de “cubierta” y que sea posible, por lo tanto, establecer para ella unos márgenes distintos a los del resto del documento, debemos asignar a la opción «`titlepage`» el valor «`firstiscover`». Pero si queremos tener

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

conjuntamente página de título y página de cubierta, podemos generar esta última mediante el comando «`\extratitle`» cuyo formato es:

```
\extratitle{Texto de la cubierta}
```

El texto recibido como argumento por «`\extratitle`» puede constar de varios párrafos y no es formateado de ninguna manera especial. El autor debe pues introducir en el texto los comandos necesarios para obtener el formato que desee.

Otra utilidad adicional que incorpora KOMA-Script a los títulos, es la posibilidad de generar algún tipo de texto para el reverso de la página de título. Ello se consigue mediante los siguientes dos comandos:

```
\uppertitleback{Texto de la zona superior del reverso}  
\lowertitleback{Texto de la zona inferior del reverso}
```

Contenido del título. La información de que consta el título se distribuye en siete campos, cada uno de los cuales se introduce mediante un comando distinto, y es formateado de manera diferente por KOMA-Script al generar el título. Los campos, expresados en el mismo orden en el que se imprimen son los siguientes:

```
\titlehead  
\subject  
\title  
\subtitle  
\author  
\date  
\publishers  
\dedication
```

Dejando de lado los campos «`\title`», «`\author`» y «`\date`», que existen en los documentos estándar de L^AT_EX, expondré muy sucintamente mi apreciación personal sobre la función de cada uno de estos elementos. Dicha apreciación la baso, por supuesto, en el nombre del elemento, pero también en el orden en el que se imprimen y en el formato que reciben por defecto.

titlehead: Este elemento está pensado para introducir información adicional que ayude a enmarcar el resto de la información del título; como, por ejemplo, el nombre de la institución donde se ha confeccionado el documento, o el título del congreso en el que este se presenta, etc.

subject: A pesar de que el término “*subject*” sugiere que este elemento está pensado para recoger el tema o la materia sobre la que versa el documento, dado que KOMA-Script lo coloca *antes* del título propiamente dicho, pienso que posiblemente con él lo que se pretende más bien es indicar el tipo de texto que viene a continuación: una charla o disertación, una comunicación, un informe, etc.

subtitle: El nombre de este elemento no parece necesitar aclaración alguna.

publishers: Por su nombre se refiere a los editores, pero supongo que también puede servir para los patrocinadores del documento, o para la institución en la que trabaja el autor, etc.

dedication: Como su propio nombre indica, se supone que este campo contendrá una o varias frases de dedicatoria o agradecimiento. Cuando «`titlepage`» está establecido a «`true`», este campo se imprime en una página independiente; y en documentos a doble página, en una página impar.

Cada uno de estos elementos recibe un formateo por defecto por KOMA- Script. Podemos cambiarlo mediante los procedimientos explicados en la sección 3.7.

El entorno abstract. Para este entorno, que se suele incluir entre el título y el índice, KOMA- Script ofrece una opción, llamada igual que él, que indica si deberá o no generarse un título que diga que ese texto es un *abstract*:

Opción: abstract
<ul style="list-style-type: none">■ <i>Tipo de valor:</i> Si/No*■ <i>Valor por defecto:</i> false■ <i>Si no se le asigna un valor:</i> true

Esta opción, por otra parte, sólo funciona en documentos del tipo `scrartcl`, pues entienden los autores de KOMA- Script que en libros e informes no es correcto un *abstract* inicial sino que, se es precisa alguna aclaración sobre el contenido del texto, a ello debe dedicarse el capítulo preliminar.

3.5.2. Índice general del documento

¿Está usted cansado de tener que indicar en su documento expresamente que desea una entrada en el índice general para la bibliografía, o para la lista de figuras? ¡Enhorabuena! En los documentos de KOMA- Script el índice general es altamente configurable y estas tareas pueden automatizarse. La configuración descansa en el valor de la opción «`toc`»:

Opción: toc
<ul style="list-style-type: none">■ <i>Tipo de valor:</i> Palabra simbólica■ <i>Valor por defecto:</i> «<code>indent</code>», «<code>nobib</code>», «<code>noidx</code>», «<code>nolistof</code>»■ <i>Si no se le asigna un valor:</i> Los valores por defecto

«`toc`» es una de las raras opciones de KOMA- Script que admiten varios valores, lo que implica que, para fijar su valor al cargar el tipo de documento, hay que invocarla varias veces. Por ejemplo:

```
\documentclass[toc=idx,toc=bib]{scrbook}
```

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

Para cambiar su valor con posterioridad podemos usar «`\KOMAOpcion`», que permite asignar varios valores a una sola opción. Por ejemplo

```
\documentclass{scrbook}
\KOMAOpcion{toc}{idx, bib}
```

Los siguientes valores «`toc`» afectan al aspecto del índice general:

graduated, indented, indent Todas estas palabras simbólicas son equivalentes. Indican que el índice de contenido se construya al modo habitual: indicando las relaciones jerárquicas mediante los sangrados de las distintas entradas.

flat, left Estos dos valores provocan que el índice se construya como una tabla con tres columnas. La primera columna recoge el número de la entrada, la segunda el texto de la entrada, y la tercera el número de página. En este tipo de índice no hay indicación visual de las relaciones jerárquicas entre las distintas entradas, por lo que puede ser adecuado cuando todas las entradas tengan el mismo nivel jerárquico.

El resto de los valores posibles de «`toc`» afecta a si en el índice se incluirá o no una entrada para la bibliografía, para las listas de objetos flotantes y para el propio índice. Para cada uno de estos elementos se ofrecen tres posibilidades, cada una de las cuales, a su vez, está representada por más de una palabra simbólica, lo que hace verdaderamente engorroso exponer este punto. Las posibilidades son las siguientes:

1. No incluir el elemento en el índice. Esto se indica mediante los valores «`nobibliography`» o «`nobib`», para la bibliografía, «`noindex`» o «`noidx`» para el índice de contenido, y «`nolistof`» para las listas de objetos flotantes. Este es el valor por defecto para los tres tipos de elementos.
2. Incluir el elemento en el índice, pero hacerlo sin que el mismo reciba en él una numeración, dejando así claro que no es un capítulo o sección. Esto se obtiene con los valores «`bibliography`» o «`bib`» para la bibliografía, «`index`» o «`idx`» para el índice de contenido, y «`listof`» para las listas de objetos flotantes.
3. incluir el elemento en el índice y numerarlo como si fuera una sección más. Esto sólo es posible para la bibliografía y para las listas de objetos flotantes. Las palabras que representan este valor son las mismas que las del apartado anterior, uniendo a ellas el texto “numbered” al principio o al final. Así, si el valor para recoger en el índice las listas de objetos flotantes es «`listof`», los valores que harán que estas listas sean numeradas serán «`numberedlistof`» y «`listofnumbered`». Para la bibliografía, por lo tanto habrá hasta cuatro nombres posibles: «`numberedbibliography`», «`bibliographynumbered`», «`numberedbib`» y «`bibnumbered`».

También es posible modificar los valores de formato que, por defecto, KOMA-Script asigna a las distintas entradas del índice. Ello se puede hacer por el procedimiento explicado en la sección 3.7.

3.6. El cuerpo del documento

3.6.1. Espaciado vertical entre párrafos

Un recurso tipográfico bastante extendido es el de resaltar visualmente los distintos párrafos de una página. Ello suele hacer principalmente de dos maneras: indentando ligeramente la primera línea de cada párrafo, distancia que es controlada por «`\parindent`», y aumentando levemente el espaciado vertical entre los párrafos, el cual depende de la longitud elástica «`\parskip`».

KOMA- Script, partiendo de la idea de que usar simultáneamente ambos procedimientos es redundante, propone al usuario elegir entre uno de ellos. Por defecto se activa la indentación y se desactiva el espaciado vertical. Pero esto se puede cambiar mediante la opción «`\parskip`»:

Opción: `\parskip`

- *Tipo de valor:* Si/No* o Palabra simbólica
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Como ambos procedimientos de destacar los párrafos están coordinados, si se activa «`\parskip`» se desactivará la indentación de la primera línea, y al revés: si se desactiva «`\parskip`» se reactivará la indentación de la primera línea.

Además de los valores propios de las opciones de activación, «`\parskip`» puede recibir los siguientes valores adicionales:

never Indica que no se desea espacio vertical extra bajo ninguna condición. Ni siquiera cuando este espacio sea necesario para el ajuste vertical que provoca «`\flushbottom`» (que no explico por no tratarse de un comando de KOMA- Script).

full Indica que se debe activar «`\parskip`» y que el espacio vertical entre párrafos debe ser de una línea.

half Indica que se debe activar «`\parskip`» y que el espacio vertical entre párrafos debe ser de media línea.

Cuando «`\parskip`» está activado, el cambio entre párrafos se indica por el espacio vertical extra. Pero este procedimiento no siempre permite ver con claridad cuándo acaba un párrafo y empieza el próximo. Concretamente, si la última línea del párrafo coincide con la última línea de la página, y además, da la casualidad de que esa última línea llega hasta el borde derecho del párrafo, el lector no podrá saber con certeza que ahí termina un párrafo.

Por ello la activación del espaciado vertical para indicar la separación de párrafos va unida a otra función: cuando «`\parskip`» está activado KOMA- Script se asegura de que la última línea del párrafo sea, al menos, 1 *em* más corta que las restantes. Este comportamiento por defecto podemos alterarlo añadiendo a los valores «`full`» y «`half`» un carácter adicional que puede ser «`*`», «`+`» o «`-`». El significado de estos caracteres es el siguiente:

- El carácter «`*`» («`full*`», «`half*`») garantiza que el espacio en blanco que quede al final

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

de la última línea mida, al menos, una cuarta parte de la longitud de la línea.

- El carácter «+» («full+», «half+») se asegura de que el espacio en blanco a la derecha de la última línea mida, al menos, un tercio de la longitud de la misma.
- El carácter «-» («full-», «half-») indica a KOMA-Script que se despreocupe por la longitud de la última línea del párrafo. No se tomará ninguna medida para asegurarse de que esta línea vaya a ser más corta que las demás.

Sumando todo lo anterior, resulta que los valores posibles para la opción «parskip», además de los propios de las opciones Si/No, son los siguientes: «never», «half», «half+», «half*», «half-», «full», «full+», «full*» y «full*».

Nota: Es importante no confundir la opción «parskip» y la longitud «\parskip». Esta última sigue existiendo en los documentos de KOMA-Script, y aunque su valor se establece por defecto a partir del valor de la opción «parskip», podemos introducirle manualmente otro valor *después* de haber fijado la opción «parskip». Y así, por ejemplo, si tras indicar «parskip=false» señalamos que «\parskip=3pt», tendremos activada la indentación de la primera línea de cada párrafo (porque «parskip=false») y un espacio vertical extra de 3 puntos (porque «\parskip=3pt»).

También podemos, por supuesto, hacer lo mismo con la longitud «\parindent».

3.6.2. Notas al pie

Notas consecutivas. Cuando en un documento se insertan dos notas consecutivas, al juntarse el número de la primera nota con el de la segunda, se produce la sensación de que se trata de un solo número y de una sola nota. A continuación insertaré dos notas para demostrar lo que quiero decir²³. Se han insertado dos notas, la 1 y la 2, pero quien lea el texto lo que verá es una llamada a la nota «12». Para evitar este efecto KOMA-Script incluye la opción «footnotes»:

Opción: footnotes

- *Tipo de valor:* Palabra simbólica
- *Valor por defecto:* nomultiple
- *Si no se le asigna un valor:* multiple

Los valores posibles para esta opción son «multiple» y «nomultiple». Con el valor «multiple» se inserta entre ambas referencias la cadena de texto que se haya definido en el comando «\multfootsep», que por defecto es una coma.

Lo expuesto es, más o menos, lo que afirma la documentación de KOMA-Script. Sin embargo en mis pruebas esta opción no ha producido ningún resultado apreciable.

Pluralidad de referencias a una misma nota Para incluir varias referencias a la misma nota en un texto, en L^AT_EX habría que usar varias veces el comando «\footnotemark» y luego escribir el contenido de la nota con «\footnoteref». KOMA-Script implementa el comando «\footref» que proporciona un procedimiento más sencillo. Su formato es:

² Esta es la nota 1.

³ Esta es la nota 2.

```
\footref{Etiqueta}
```

El procedimiento a seguir consistiría en, la primera vez que se pretende hacer referencia a la nota, escribirla del modo normal, con «\footnote» y, dentro del texto de la nota, añadir una etiqueta con «\label». Tras ello, el resto de las referencias se harían no con «\footnote» sino con «\footref» indicando, como argumento del comando, la etiqueta que representa a la nota a la que nos queremos referir.

Ojo: Este procedimiento simplifica la tarea, pero no es perfecto pues es posible que al compilar el documento, alguna de las referencias a la nota en cuestión caiga en una página distinta de la página en la que se encuentra el texto de la nota.

Control del formato de la nota. En una nota a pie de página distinguimos entre la *marca* y el contenido. La marca es el número, letra o símbolo que la identifica en la página. La marca aparece en dos lugares: en el punto del cuerpo del texto en el que se hace la llamada a la nota (la *referencia*), y en el pie de página, al empezar la nota, justo antes de su contenido.

KOMA- Script nos permite controlar el formato de la marca, y la indentación del texto de la nota con relación a la marca⁴. Los comandos con los que ello se consigue son:

```
\thefootnotemark  
\deffootnotemark{Definición}  
\deffootnote[Ancho de la marca]{indentación}{parindent}{definición}
```

El primer comando se limita a devolver el contenido de la marca, sin ningún tipo de formato (a diferencia de «\thefootnote» que devuelve el contenido de la marca, formateado). Este valor se usa en los dos comandos siguientes.

Mediante el comando «\deffootnotemark» podemos indicar el formato en el que queremos que se muestren las referencias que, dentro del texto, se hacen a las notas. Por ejemplo: el siguiente comando hará que las notas se muestren, en el texto, en superíndice y negrita.

```
\deffootnotemark{\textbf{\textsuperscript{\thefootnotemark}}}
```

Por último, el comando «\deffootnote» nos permite controlar varios aspectos de la nota tal y como se muestra al pie de página. En concreto:

- El argumento opcional indica la anchura de la caja en la que se imprimirá la marca de la nota en el pie de página.
- El argumento “indentación” indica la indentación de la segunda y sucesivas líneas del texto de la nota propiamente dicho.
- El argumento “parindent” indica el espacio adicional en el que se debe indentar la primera línea del segundo y posteriores párrafos de la nota.
- Por último, el argumento “definición” indica el formato que hay que aplicar *a la marca de la nota* que se imprime al principio de la misma al pie de la página.

⁴ Otros aspectos del formato de la marca se pueden controlar mediante los procedimientos descritos en la sección 3.7

Para entenderlo, lo mejor es examinar el valor que «`\deffootnote`» tiene por defecto en KOMA- Script:

```
\deffootnote[1em]{1.5em}{1em}{\textsuperscript{\thefootnotemark}}
```

Si nos fijamos se está indicando, que como anchura para la marca de la nota se reserve 1 *em*, que, no obstante, la indentación de las líneas de la nota, a partir de la segunda sea de 1.5 *ems* y, en las líneas que empiecen párrafo se añada un *em* adicional; y finalmente que la marca se escriba en superíndice.

Esta definición, que es la que se aplica por defecto en KOMA- Script provoca que el texto de la primera línea de la nota aparezca ligeramente desplazado a la izquierda con respecto a la segunda línea. Ello es porque la primera línea del texto empieza justo detrás de la marca, la cual ocupa una anchura de 1 *em* y, sin embargo, las líneas segunda y siguientes se indentan en 1.5 *ems*. Para evitar ese ligero desplazamiento, bastaría con alterar el argumento opcional y hacerlo valor 1.5 *ems*. Y para conseguir que la marca se alinee a la izquierda:

```
\deffootnote{1.5em}{1em}{\makebox[1.5em][l]{\thefootnotemark}}
```

Línea separadora entre el texto y las notas. Podemos también controlar el grosor y la longitud de la línea que separa las notas del texto. Para ello se implementa el siguiente comando:

```
setfootnoterule[Grosor]Longitud
```

También podemos cambiar el color de la línea, mediante los procedimientos descritos en la sección 3.7.

3.6.3. Partes, capítulos, secciones

Alineación de los títulos. Los siguientes comandos controlan la alineación de los títulos de las unidades estructurales:

```
\raggedsection  
\raggedchapter  
\raggedpart
```

«`\raggedsection`» afecta, en principio, a todas las unidades estructurales de capítulo para abajo. «`\raggedchapter`» sólo afecta a los capítulos, y «`\raggedpart`» a las partes.

Estos comandos pueden ser modificados mediante «`\renewcommand`». Y así, por ejemplo, si queremos que los títulos de capítulos y secciones aparezcan centrados, en el preámbulo de nuestro documento podemos escribir:

```
\renewcommand{\raggedsection}{\centering{}}
```

Tipo de página en el que se inician los capítulos. Por defecto, en los libros impresos a doble cara, los capítulos se inician siempre en página impar. Este funcionamiento por defecto puede ser alterado mediante la opción «`open`» que sólo se admite en los libros e informes, no en los artículos:

Opción: open

- *Tipo de valor:* Palabra simbólica
- *Valor por defecto:* right
- *Si no se le asigna un valor:* right

Los valores posibles para esta opción son:

any Los capítulos se inician en la página siguiente, con independencia de que ésta sea par o impar.

left Los capítulos se inician necesariamente en una página par.

right Los capítulos se inician necesariamente en una página impar.

El valor de esta opción, por otra parte, afecta al funcionamiento del comando «`\cleardoublepage`» (de L^AT_EX estándar) así como al de los comandos (explicados, aunque muy poco, en la sección 3.6.4.) «`\cleardoublepageusingstyle`», «`\cleardoubleemptypage`», «`\cleardoubleplainpage`» y «`\cleardoublestandardpage`» (implementados por KOMA-Script).

A partir de la versión 3 de KOMA-Script, la utilidad de esta opción se ha incorporado (también) a la opción «`headings`» que se explica más adelante, en esta misma sección.

Prefijo de los capítulos y apéndices. En los documentos estándar de L^AT_EX, el título de los capítulos se formatea en dos líneas: En la primera se contiene la palabra “Capítulo”⁵ y el número del capítulo. En la segunda se contiene el título propiamente dicho. Lo mismo ocurre en los apéndices, salvo que en ellos la palabra que precede al número de apéndice es “Apéndice”. En KOMA-Script, por defecto, se suprime la palabra “capítulo” o “Apéndice” y el resto se imprime en una sola línea.

Podemos volver al funcionamiento de las clases estándar de L^AT_EX alterando el valor de las opciones «`chapterprefix`» y «`appendixprefix`»:

Opción: chapterprefix

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Opción: appendixprefix

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

El valor de estas opciones afecta no sólo al título del capítulo o apéndice, sino también al texto que será automáticamente enviado al encabezado al cambiar de capítulo.

⁵ En realidad lo que se imprime es el texto definido en el comando «`\chaptername`». La opción «`spanish`» de «`babel`» es la que se ocupa de que ese texto sea “Capítulo”.

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

A partir de la versión 3 de KOMA-Script, la utilidad de esta opción se ha incorporado (también) a la opción «headings» que se explica a continuación.

headings: control de determinados aspectos formales de los títulos La opción «headings», implementada en la versión 3.0 de KOMA-Script auna la funcionalidad de las dos opciones anteriores, y, además permite controlar otros aspectos como el tamaño del título. Su formato es:

Opción: headings
<ul style="list-style-type: none">■ <i>Tipo de valor:</i> Palabra simbólica■ <i>Valor por defecto:</i> Múltiples valores■ <i>Si no se le asigna un valor:</i> Se aplican los valores por defecto

Esta opción (que es de las que admite varios valores simultáneamente) permite controlar los siguientes aspectos de los títulos:

Tipo de página en el que se inician los capítulos. Es decir: exactamente lo mismo que controla la opción «open». Los valores de «headings» que se refieren a esta cuestión son «openany», «openleft» y «openright» y cada uno de ellos significa lo mismo que el valor equivalente de «open».

Hay, no obstante, una importante diferencia entre «open» y «headings»: mientras la primera afecta exclusivamente a los capítulos, «headings» afecta también a las partes, al índice y al comando «\backmatter».

Prefijo de los capítulos y apéndices. O sea: exactamente lo mismo que controla la opción «chapterprefix». Los valores que admite en este punto esta opción son una combinación de las palabras “appendix” o “chapter” y “oneline”, “withoutprefix”, “withoutprefixline”, “twoline” “withprefix” o “withprefixline”. De manera que para lo que sustancialmente admite sólo dos valores, si o no, esta opción llega a admitir 8 palabras simbólicas. No estoy seguro de que este procedimiento, usado por KOMA-Script con frecuencia, realmente facilite su aprendizaje. Desde luego, para la escritura de guías, es un engorro.

Resumiendo: Hacen que el título del capítulo se imprima en dos líneas los valores *twolinechapter*, *chapterprefix*, *chapterwithprefix* y *chapterwithprefixline*. Por el contrario los valores *onelinechapter*, *nochapterprefix*, *chapterwithoutprefix* y *chapterwithoutprefixline* provocan el efecto contrario. Estos mismos comandos, cambiando “chapter” por “appendix” afectan a los apéndices.

Tamaño de la fuente y del espaciado vertical del título. Aquí los valores posibles son «big», «normal» y «small». Con big (valor por defecto) los títulos de los capítulos se generan con una fuente muy grande, y se separa el título del texto con bastante espacio vertical. «normal» reduce la fuente y el espacio vertical a unos valores medios y «small» los reduce aún más.

De acuerdo con mis pruebas, «big» y «normal» sólo afectan a las partes y a los capítulos. «small» afecta también a otras unidades estructurales.

Relación entre el título, el índice y el encabezado. En L^AT_EX las unidades estructurales pueden recibir un argumento opcional que sustituye al título de la unidad en el índice y en

el encabezado. La opción «headings» nos permite precisar si queremos enviar dicho argumento opcional sólo al índice, sólo al encabezado, o a ambos lugares. Los valores posibles son los siguientes:

- **optiontohead:** El argumento opcional de las unidades estructurales será enviado sólo al encabezado, no al índice.
- **optiontoheadandtoc, optiontotocandhead:** El argumento opcional de las unidades estructurales será enviado al encabezado y al índice.
- **optiontotoc:** El argumento opcional de las unidades estructurales será enviado al índice, pero no al encabezado.

Secciones no numeradas, pero enviadas al índice o al encabezado. Todas las unidades estructurales de L^AT_EX admiten dos versiones: la versión normal y una versión con asterisco que implica que dicha sección no debe numerarse ni aparecer en el índice o enviarse al encabezado. Para el caso de que deseemos unidades estructurales no numeradas, pero sí reflejadas en el índice y en el encabezado KOMA-Script implementa los siguientes comandos:

```

\addpart[Versión corta]{Título}
\addchap[Versión corta]{Título}
\addsect[Versión corta]{Título}

\addpart*{Título}
\addchap*{Título}
\addsec*{Título}

```

Los comandos sin asterisco generan, respectivamente, partes, capítulos y secciones no numeradas, pero que tendrán su reflejo en el índice y, si el estilo de página lo exige, se mostrarán en el encabezado. La versión con asterisco de estos comandos es similar a la versión con asterisco de los comandos clásicos («\part*», «\chapter*» y «\section*» con la única diferencia de que, aunque el título no sea enviado al encabezado, sí se borrará el encabezado vigente hasta entonces.

Numeración de los títulos La opción «numbers» controla si la numeración de las unidades estructurales debe o no terminar con un número:

Opción: numbers

- *Tipo de valor:* Palabra simbólica
- *Valor por defecto:* autoendperiod
- *Si no se le asigna un valor:* autoendperiod

Admite tres valores: Con «endperiod» la numeración automática terminará con un punto; con «noendperiod» no habrá punto al final de la secuencia y con «autoendperiod» se aplicarán las reglas del documento DUDEN sobre la corrección tipográfica al escribir en alemán (por algo KOMA-Script ha sido hecho en Alemania).

Para documentos en español debe tenerse en cuenta que la opción «spanish» de «babel» anula los efectos de la misma y aplica las reglas tipográficas españolas.

Minisecciones. KOMA-Script denomina “minisecciones” a una línea de texto, que, al igual que las unidades estructurales, va formateada en negrita, y vinculada al párrafo siguiente, de tal manera que no puede haber entre ambos un salto de página, y que, sin embargo, no es una verdadera unidad estructural, no aparece en el índice ni en el encabezado del documento.

Estas *minisecciones* se introducen con el comando «\minisec»:

```
\minisec{Título}
```

Preámbulos. En libros e informes es posible añadir un fragmento de texto que cumpla la función de hacer de preámbulo a una parte o a un capítulo. Este texto se imprimiría inmediatamente encima o debajo del título de la parte o capítulo, e incluso es posible generar un preámbulo para imprimir *antes* y otro para incluir *después*.

Esto es especialmente útil en las partes de los libros, porque aunque por defecto en estos documentos se reserva toda una página para el título de la parte, el preámbulo se imprime en dicha página. También cuando el documento es a doble columna, porque el preámbulo, al igual que el título de los capítulos, se imprime en una sola columna.

Los comandos que generan los preámbulos son

```
\setpartpreamble[Posición][Anchura]{Preámbulo}
\setchapterpreamble[Posición][Anchura]{Preámbulo}
```

Donde los argumentos tienen el siguiente significado:

Posición: es una combinación de dos letras. La primera representa si el preámbulo debe figurar por encima del título («o») o por debajo («u»), y la segunda representa si la alineación del preámbulo debe ser a la izquierda («l»), a la derecha («r») o centrada («c»). Si no se indica nada el preámbulo se coloca por debajo del título y centrado.

En el caso de los capítulos la presencia de un preámbulo no hará que el título se desplace hacia abajo, ni siquiera aunque el preámbulo se vaya a colocar por encima de él. Debemos por lo tanto asegurarnos de que su extensión no sea tanta como para no caber en el espacio que queda por encima del título.

Anchura: Indica la anchura de la caja en la que se insertará el preámbulo. Si no se incluye este argumento, se usará toda la anchura de la línea.

Preámbulo: Es el contenido del preámbulo. Puede constar de varios párrafos e incluir en su interior entornos.

3.6.4. Estilos de página, encabezados y pies

En L^AT_EX el contenido y forma del encabezado y del pie de página viene determinado por los llamados *estilos de página*. KOMA-Script reconoce los mismos estilos de página que L^AT_EX

estándar, si bien en alguno de ellos introduce pequeños retoques tipográficos que afectan, sobre todo, a la alineación de los elementos que se muestran en el encabezado y en el pie de página, en documentos a doble cara. Implementa además un procedimiento que permite cambiar con relativa facilidad el formato por defecto de los elementos del encabezado y del pie (véase la sección 3.7).

Aparte de eso, se introducen algunas variables que afectan directamente al encabezado y al pie (con independencia del estilo de página que se esté usando), así como un conjunto de comandos que flexibilizan mucho el uso de las hojas de estilo. Por último, mediante el paquete «`scrlayer-scrpage`» es posible generar nuestros propios estilos de página, si bien ese aspecto no será tratado en esta guía.

Opciones que afectan al contenido de los encabezados y pies de página. Además de las opciones «`headlines`», «`headheight`», «`footlines`» y «`footheight`», que ya hemos visto en la sección 3.3.3, se añaden las siguientes opciones:

Opción: `headsepline`

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Opción: `footsepline`

- *Tipo de valor:* Si/No*
- *Valor por defecto:* false
- *Si no se le asigna un valor:* true

Estas opciones determinan si existirá o no una línea de separación entre el cuerpo del texto y el encabezado o el pie de página.

Comandos para fijar el estilo de ciertas páginas especiales. Hay determinadas páginas en las que no interesa aplicar el estilo de página del resto del documento. \LaTeX dispone del comando «`\thispagestyle`» para ello. Pero KOMA-Script da un paso más y genera un conjunto de comandos que permiten determinar el estilo de ciertas páginas en las que habitualmente se invoca «`\thispagestyle`». Se trata de los siguientes comandos:

`\titlepagestyle` Determina el estilo a usar en la página en la que aparezca el título, cuando la opción «`titlepage`» está desactivada, es decir: cuando no se reserva una página entera para el título.

`\partpagestyle` Determina el estilo a usar en la página generada por la unidad estructural «`\part`».

`\chapterpagestyle` Determina el estilo a utilizar en la primera página de un capítulo. No está disponible en documentos del tipo «`scrartcl`».

* Sobre los valores posibles para las opciones del tipo Si/No, véase la tabla 1 en la página 6.

\indexpagestyle Determina el estilo a utilizar en la primera página del índice de contenido.

No está disponible en documentos del tipo «`scrartcl`».

Estos comandos son en realidad variables que almacenan el nombre de la página de estilo a usar en el contexto al que cada uno de ellos se refiere. Por lo tanto hay que usar «`\renewcommand`» para modificar su contenido. Y así, por ejemplo, si deseo que en la página en la que aparece un capítulo no se imprima el número de página, deberé ejecutar el comando:

```
\renewcommand{chapterpagestyle}{empty}
```

Páginas de relleno y comportamiento de «`\cleardoublepage`» En documentos impresos a doble cara hay elementos que siempre deben empezar en página impar como las partes, capítulos, índice de contenido, etc. Para conseguir esto en ocasiones el compilador tiene que insertar una página de relleno, la cual, en los documentos generados con las clases estándar, aunque está vacío, sigue teniendo el encabezado y el pie de página propio del estilo que se esté usando, lo cual provoca un efecto, en mi opinión, muy feo. Una página en blanco debería estar totalmente en blanco.

KOMA- Script dispone de dos opciones que evitan ese resultado:

Opción: `cleardoublepage`

- *Tipo de valor:* El nombre de un estilo o «`current`»
- *Valor por defecto:* `current`
- *Si no se le asigna un valor:* `current`

Con esta opción podemos indicar qué estilo de página se debe aplicar a las páginas de relleno generadas por los dobles saltos de página. De modo que la indicación de «`cleardoublepage=empty`» evitaría el problema que se acaba de mencionar.

Si se desea que en las páginas de relleno se siga aplicando el estilo que en ese instante esté activo (funcionamiento por defecto) debemos indicar como valor para esta opción «`current`».

Junto a esta opción, KOMA- Script aporta un grupo de comandos dirigidos a controlar el funcionamiento de «`\cleardoublepage`». Se trata de:

```
\cleardoubleemptypage  
\cleardoubleplainpage  
\cleardoublestandardpage  
\cleardoublepageusing{Estilo}  
  
\cleardoubleoddemptypage  
\cleardoubleoddplainpage  
\cleardoubleoddstandardpage  
\cleardoubleoddusing{Estilo}  
  
\cleardoubleevenemptypage  
\cleardoubleevenplainpage  
\cleardoubleevenstandardpage  
\cleardoubleevenusing{Estilo}
```

Sobre estos comandos, poco puedo decir, puesto que no termino de entender qué aportan respecto a la opción «`\cleardoublepage`» ni en qué se diferencian unos de otros⁶

3.6.5. Tablas y figuras flotantes

3.7. Modificación del formato aplicado por defecto a los distintos elementos del documento

4. Otros comandos y entornos que no existen en \LaTeX estándar

Además de los comandos vistos en las secciones anteriores, KOMA-Script incorpora los siguientes comandos:

4.1. Citas o dichos

En muchos escritos se acostumbra a poner una pequeña frase o cita al inicio o al final de un capítulo o sección. Esa frase suele formatearse de una manera especial y es relativamente habitual que se alinee a la derecha.

KOMA-Script implementa varias herramientas para tratar con este tipo de textos a los que globalmente denomina *dicta* que es —supongo— el plural en latín de *dictum* = *dicho*.

Los comandos que gestionan estos “dichos” son los siguientes:

<pre>\dictum[Autor]{Texto} \dictumwidth \dictumauthorformat{Autor} \dictumrule \raggeddictum \raggeddictumtext \raggeddictumauthor</pre>
--

De todos ellos el primero es el comando principal, responsable de generar la cita. El resto de los comandos se ocupa de precisar su aspecto.

La cita se inserta en una caja de anchura igual a «`\dictumwidth`» (por defecto un tercio de la línea). La caja se alinea según se indique en «`\raggeddictum`», que por defecto es a la derecha («`\reggedleft`»). El texto del interior de la caja se alinea en la dirección que indique «`\raggeddictumtext`»; por defecto a la izquierda («`\raggedright`»). Si se ha indicado el autor, bajo la caja se imprime una línea definida como objeto vertical por el «`\dictumrule`», y a

⁶ Admito que no termino de entender el funcionamiento de estos comandos. Tal y como yo lo entiendo, el primer grupo de comandos se aseguran de aplicar una determinada hoja de estilo (empty, plain, la hoja activa o la que se indique expresamente) a la posible hoja de relleno que pueda generar «`\cleardoublepage`». Pero los otros dos grupos de comandos... se me escapa su utilidad. Los comandos que llevan “odd” en su nombre, parecen referirse a páginas de relleno impares... pero eso no es posible, porque las páginas de relleno siempre son pares. La documentación de KOMA-Script además dice que estos comandos se aplican a las páginas “izquierdas”, que son las pares. Pero entonces... ¿Hacen lo mismo que el grupo anterior? ¿Y qué pasa con el tercer grupo?

continuación se imprime el autor, alineado en la dirección que indique «`\raggeddictumauthor`» (por defecto a la derecha). Antes de imprimir el autor su nombre es pasado como argumento al comando «`\dictumauthorformat`» que puede formatearlo, o rodearlo entre paréntesis...

La modificación de cualquiera de los comandos que controla la apariencia del dicho, debe ser hecha con «`\renewcommand`». Así, por ejemplo, si nuestra cita es relativamente larga y deseamos permitirle que ocupe la mitad de la línea, deberíamos ejecutar:

```
\renewcommand*{\dictumwidth}{.5\textwidth}
```

4.2. Descripciones alternativas a «`description`»: El entorno `labeling`

Se trata de un entorno de lista pensado para descripciones. Su peculiaridad está en que en él podemos controlar la separación entre la etiqueta y su descripción, así como la indentación de los elementos del entorno. Su formato es el siguiente:

```
\begin{labeling}[delimitador]{Patrón de indentación}
\item [Etiqueta]
...
\end{labeling}
```

Para explicar los argumentos de este entorno, en lugar del entorno «`description`», que es el que suelo usar, usaré el propio entorno «`labeling`» con la siguiente declaración:

```
\begin{labeling}[ --- ]{Tararí}
```

delimitador — Es un argumento opcional. Se trata de un texto que se colocará entre la etiqueta y el contenido de los elementos de la lista.

Patrón de indentación — Se trata de un texto arbitrario, cuya anchura se usará para medir la indentación de los elementos del entorno.

4.3. Modificar los márgenes de un fragmento de texto: El entorno `addmargin`

Los entornos «`quote`» y «`quotation`» incrementan los márgenes izquierdo y derecho, pero lo hacen en una medida fija que el autor no puede modificar. Mucho más flexibles son, por lo tanto, los entornos «`addmargin`» y «`addmargin*`», implementados por KOMA-Script.

Su formato es el siguiente

```
\begin{addmargin}[Indentación izquierda]{Indentación}
...
\end{addmargin}

\begin{addmargin*}[Indentación interior]{Indentación}
...
\end{addmargin*}
```

Ambos entornos reciben un argumento opcional y otro obligatorio. Si sólo existe el argumento obligatorio, éste se interpreta como la medida en que se indentarán ambos márgenes. Si se recibe también el argumento opcional, cambia la interpretación del argumento obligatorio.

- En «`addmargin`» se tomarán los argumentos opcional y obligatorio, respectivamente, como la medida en que hay que aumentar la indentación izquierda y derecha.
- En «`addmargin*`» se tomarán los argumentos opcional y obligatorio, respectivamente, como la medida en que hay que aumentar la indentación interior y la indentación exterior.

Ambos argumentos son, por otra parte, medidas de \LaTeX y requieren en consecuencia son se indique la unidad de medida. El argumento obligatorio no puede quedar vacío. Si no se desea indentación hay que indicar expresamente el valor de «0» seguido de alguna unidad de medida. Por ejemplo: «0pt». Se admiten también valores negativos, lo que nos da un total control sobre los márgenes deseados para el fragmento de texto encerrado en estos entornos.

«`addmargin*`» sólo tiene sentido en documentos donde la opción «`twoside`» esté activada. Si se usa en documentos a una sola cara, funciona exactamente igual que «`addmargin`».

4.4. Otros comandos

Texto en subíndice. \LaTeX define el comando «`\textsuperscript`» para escribir texto en superíndice, pero no incluye un comando similar para el texto en subíndice. Este se puede obtener, claro está, en modo matemático usando el carácter reservado `_`, pero ello tiene el inconveniente de que, al tener que activarse el modo matemático, se usa la fuente asociada a él. KOMA-Script remedia esta omisión definiendo el comando «`\textsubscript`» para este propósito, cuyo formato es:

```
\textsubscript{Texto}
```

Detección de páginas pares e impares. En determinadas ocasiones, sobre todo en documentos impresos a doble cara, podemos necesitar saber si un punto concreto del documento se encuentra ubicado en una página par (izquierda) o en una página impar (derecha). Para ello KOMA-Script implementa el siguiente comando:

```
\ifthispageodd{Verdadero}{Falso}
```

El comando recibe dos argumentos. Si la página en la que se encuentra resulta ser impar, ejecutará o escribirá el primer argumento. En el caso contrario, si la página era par, se ejecutará o escribirá el contenido del segundo argumento.

Para que los resultados de este comando sean fiables, son precisas al menos dos compilaciones.