

Elementos básicos^(*) de texto

en L^AT_EX para docentes

Álvaro Crego Deán

Índice

1. Estrutura dun documento xenérico	2
1.1. Seccións e subseccións	3
2. Forma do texto	3
2.1. Parágrafos, saltos de carro e de páxina	3
2.2. Marxes, indentados e interliñados	4
2.3. Texto a columnas	4
2.4. Alíñar texto	5
2.5. Espazos en branco	5
3. Caracteres especiais	5
3.1. Caracteres para comentar e outros caracteres	5
3.2. Comiñas	6
3.3. Entorno matemático	6
4. Fontes e estilos texto	6
4.1. Deseño da fonte	6
4.2. Tamaño da fonte	7
4.3. Estilo da fonte	7
4.4. Texto con cor	7
5. Listaxes	8
5.1. Listaxes sen ordenar	8
5.2. Listaxes ordenadas	8
6. Imaxes	9
7. Táboas	11
8. Creación de comandos	12

(*) Este documento pretende resumir os elementos básicos de texto que poden ser de utilidade para calquer

1. Estrutura dun documento xenérico

A estrutura de calquer documento L^AT_EX componse de dúas partes. A primeira é a Cabeceira. É aquela na que lle proporcionamos ao compilador toda a información necesaria para producir o documento (tipo de documento, tamaño do papel, fonte, marxes, sangrado, paquetes...). Trátase dunha parte mecánica que se adoita copiar e pegar directamente (e non elaborar cada vez que se pretenda crear un novo documento) e que non se “traducirá” no noso documento .pdf. A segunda é o que resta: Apertura, Corpo (no que figurará todo o que si se reproducirá no .pdf) e Peche. Velaquí un exemplo de código básico:

% CABECEIRA

```
\documentclass[12pt]{article}
\usepackage[galician]{babel}           % Idioma
\usepackage[utf8x]{inputenc}          % Til e ñ directamente
\galiciandecimal{.}                   % Distingue ./, en $$
\usepackage[T1]{fontenc}               % Cambiar fonte por defecto
\usepackage{bm}                       % Letras gregas en negriña
\usepackage{graphicx}                  % Uso de imaxes
\usepackage{amsmath,amsfonts,latexsym,color,textcomp,anysize}

\marginsize{2.5cm}{2.5cm}{2cm}{2cm}   % Marxes (esq., der., sup., inf.)
\parindent=0mm                        % Sangrado
\parskip=2mm                          % Espazo entre parágrafos

\title{Título do documento}
\author{Nome do autor}
\date{12 de outubro de 1492}          % Hoxe: \today / Sen data: \empty
```

% APERTURA

```
\begin{document}
```

% CORPO

```
\maketitle
```

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero...

% PECHE

```
\end{document}
```

docente que queira comezar a empregar L^AT_EX. Nótese, por tanto, que se obviarán outras cuestións que, aínda sendo de carácter básico, son de menor utilidade para a elaboración de materiais didácticos. Así mesmo, existen outras variantes de código diferentes ás amosadas que conseguen executar boa parte das instrucións expostas

1.1. Seccións e subseccións

No caso de querer dividir o noso documento en partes ben diferenciadas, será útil empregar seccións co entorno `\section{}`. L^AT_EX numeraráas automaticamente a menos que empreguemos a variante `\section*{}`. Así mesmo, as subseccións, sub-subseccións e demais divisións escribiríanse `\subsection{}`, `\subsubsection{}` e así sucesivamente. L^AT_EX producirá un índice atendendo a estas particións. No caso de desexar facelo visible deberemos escribir a liña de código `\tableofcontents` ao comezo do Corpo do documento (véxase neste propio documento).

2. Forma do texto

2.1. Parágrafos, saltos de carro e de páxina

Para que o compilador recoñeza un cambio de parágrafo no texto é necesario deixar unha liña en branco no medio. É dicir:

Texto no parágrafo 1

Texto no parágrafo 2

Consecuentemente, se escribimos no código

Que din os rumorosos
na costa verdescente

o compilador traducirá “Que din os rumorosos na costa verdescente”. Para lograr o salto de carro (de liña) desexado sen deixar un espazo vertical en branco tamén se poden engadir ao final da primeira liña dous *backslash*:

Que din os rumorosos\\
na costa verdescente

Convén engadir o feito de que L^AT_EX está preparado para escribir por defecto os textos en modo xustificado (todas as liñas coa mesma lonxitude), mais pode darse a casualidade (moi raramente ou en texto escrito a columnas) de que o compilador teña que “cortar” algunha palabra ou que, pola contra, unha palabra “sobresaia” máis da conta ao final de liña. Nestes casos existe unha orde para obrigar a que o salto de carro se efectúe artificialmente: `\linebreak[n]`, sendo *n* un número natural entre 0 e 4 indicando o grao de importancia co que o compilador debe obedecer a esa orde. Por exemplo, coa seguinte orde lograremos que L^AT_EX curte a palabra por onde lle mandamos.

aula vir-`\linebreak[4]`tual

Por outra banda, se o que precisamos é un salto de páxina, simplemente introduciremos a orde `\newpage`.

ao longo do documento. Porén, preséntanse predominantemente aquelas de maior sinxeleza no tocante á súa

2.2. Marxes, indentados e interliñados

A estrutura do xeral do texto en canto a estas cuestións resúmese nas tres ordes que poden verse na cabeceira do código da sección 1:

```
\marginwidth{2.5cm}{2.5cm}{2cm}{2cm}    % Marxes (esq., der., sup., inf.)
\parindent=4mm                          % Sangrado (0mm se non o queremos)
\parskip=2mm                             % Espazo entre parágrafos
```

As ordes `\marginwidth{ }{ }{ }{ }`, `\parindent` e `\parskip` determinarán a lonxitude das marxes, do sangrado e do espazo entre parágrafos, respectivamente. Pode darse a casualidade de querer que un parágrafo non teña sangrado (aínda que si o teña o resto do documento), situación que solucionaremos mediante a introdución dun `\noindent` ao comezo do mesmo.

Para determinar o interliñado do texto deberemos escribir na cabeceira do documento a liña de código `\renewcommand{\baselinestretch}{ }`, introducindo entre chaves o espaciado desexado. Así, a orde

```
\renewcommand{\baselinestretch}{2}
```

provocará que o texto do documento se escriba a interliñado dobre.

2.3. Texto a columnas

O entorno máis empregado é o `\multicol`, para o cal é necesario declarar `\usepackage{multicol}` na Cabeceira do documento. A sintaxe a seguir é a seguinte:

```
\begin{multicols}{3} % entre chaves o número de columnas que se desexe.
  Texto a tres columnas.
\end{multicols}
```

Por exemplo:

O litoral galego alberga máis de cincocentas especies de macroalgas. Esta notable diversidade e abundancia non pasou desapercibida para o pobo, que acudiu nas épocas de maior penuria a elas, amais de para o tradicional uso agro-	pecuario, para consumo propio como complemento alimenticio. Non foi ata finais do s. XIX que se desenvolveu o seu aproveitamento industrial, momento no que a actividade extractiva comezou a ser regulada polo Estado. Pa-	sada a Segunda Guerra Mundial, o sector, que xa fora medrando progresivamente, experimentou unha nova expansión. Das catro empresas que se estableceron en Galicia nos anos sesenta, dúas fixérono en Ribadeo.
---	--	---

escritura.

2.4. Aliñar texto

Como xa se mencionou en 2.1, L^AT_EX escribe por defecto os textos en modo xustificado, pois considérase a forma máis “elegante”. Evidentemente, podemos cambialo. No caso de querer que unha parte do texto apareza centrado empregaremos o seguinte entorno:

```
\begin{center}
  Texto centrado...
\end{center}
```

Tamén é posible aliñar o texto á esquerda ou á dereita usando

```
\begin{flushleft} % \begin{flushright} á dereita
  Texto aliñado á esquerda...
\end{flushleft} % \end{flushright} á dereita
```

2.5. Espazos en branco

L^AT_EX interpretará do mesmo xeito `Un gato comeu ao rato` que `Un gato comeu ao    rato`, devolvendo en ambos casos o texto “Un gato comeu ao rato”. É dicir, os espazos en branco que introduzamos manualmente non serán traducidos ao documento. Os comandos para crear espazos horizontais e verticais baleiros serán `\hspace{}` e `\vspace{}`, respectivamente, aclarándose entre chaves as lonxitudes desexadas. Por exemplo:

```
\hspace{0.5cm} % Introducirá un espazo horizontal de medio centímetro.
\vspace{-3mm}  % Restará un espazo vertical de tres milímetros.
```

Son moi útiles outros comandos para axilizar o tempo de escritura ao necesitar pequenos espazos horizontais de maneira sistemática (por exemplo empregando o modo matemático) tales como `\qqquad`, `quad` ou simplemente `\`, que xerarán pequenos espazos en branco en orde decrecente de lonxitude.

3. Caracteres especiais

3.1. Caracteres para comentar e outros caracteres

L^AT_EX ten reservados algúns símbolos do teclado para empregar no propio código. É o caso de ‘#’ ou das chaves ‘}’ e ‘{’ empregados na definición de comandos. Se os tecleamos no medio do noso texto provocarán un erro de compilación. Tamén está reservado o ‘%’, que se emprega para comentar liñas de código. Se temos a necesidade de que estes símbolos se traduzan no documento .pdf, bastará escribilos cun *backslash* diante: `\#`, `\}`, `\{` e `\%`. Con isto conseguiremos que o compilador os entenda como parte do texto reproducíble no .pdf, e non como parte do código.

Comentar liñas de código (que ao levar ‘%’ diante non se traducirán) pode ser de gran utilidade para facer indicacións ou para acotar un problema de compilación cuxa orixe descoñecemos. Por iso, existen dúas maneiras a maiores para comentar código:

- Depois de seleccionar as liñas que pretendemos comentar, presionar ao mesmo tempo as teclas ‘Ctrl.’ + ‘Shift’ + ‘/’ (é a máis usada por ser a máis veloz).
- Abrir e pechar un entorno¹ dentro do cal todo que estea escrito figurará como comentado:

```
\begin{comment}
  Liñas comentadas
  ...
\end{comment}
```

3.2. Comiñas

En L^AT_EX as comiñas non responden á súa escritura habitual no teclado. Para implementar a comiña saxona deberemos empregar, como apertura, o til localizado á dereita da letra P e, como peche, o til localizado á dereita do 0. Así, o código

[...] e tal e como Sócrates pronunciou: “‘quen non pode, ao dente acode’”.

traducirase como

[...] e tal e como Sócrates pronunciou: “quen non pode, ao dente acode”.

3.3. Entorno matemático

Aínda que se adicará un documento enteiro a este entorno, é necesario saber que todo o que se escriba entre dólares ‘\$’ estará neste entorno. As letras gregas, que se escriben cun *backslash* diante, pertencen tamén ao entorno matemático. Por exemplo, α traducirase como ‘ α ’.

4. Fontes e estilos texto

4.1. Deseño da fonte

Existen dúas modalidades de cambio de fonte: local ou global.

- (a) Para cambiar a fonte localmente (dentro dunha liña ou palabra) chegará con empregar a sentencia

```
{\fontfamily{X}\selectfont texto escrito na fonte X}
```

Algunhas das fontes máis empregadas teñen por abreviatura: cmr, cmss, cmtt, cmm, cmsy, cmex... (substituír a X na sentencia de código anterior por calquera das anteriores abreviaturas).

- (b) Para cambiar globalmente (en todo o documento) a fonte debemos declarala como paque-

¹Necesítase do paquete `comment` para empregalo.

te na Cabeceira. Por exemplo, o paquete para usar Times New Roman é `\usepackage{mathptmx}`. Ademais do anterior, normalmente haberá que engadir algunha outra liña de código ao proceso, mais isto dependerá de cal sexa a fonte empregada. O recomendable é consultar o amplo catálogo <https://tug.org/FontCatalogue/> das fontes existentes, no que xa se indica como empregar cada unha delas.

4.2. Tamaño da fonte

O tamaño global da fonte xa se declara na primeira liña da Cabeceira do código. En cambio, outra maneira máis invasiva pero directa de modificalo é declaralo xusto a continuación da Apertura do documento, simplemente escribindo `\Large`, `\Huge` ou o que for mester dentro da serie de tamaños que L^AT_EX ofrece. A maiores, poderemos aplciar cambios de tamaño nos lugares do documento que desexemos mediante a seguinte sentencia:

```
{\tamañoX texto escrito a tamaño X}
```

Por exemplo,

```
{\Large O lobo} achegouse {\scriptsize en silencio} ao rabaño
```

producirá

“O lobo achegouse en silencio ao rabaño”.

L^AT_EX ofrece, de maior a menor, dez tamaños diferentes, a saber: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\LARGE`, `\huge` e `\Huge`. Para maior detalle: https://www.overleaf.com/learn/latex/Font_sizes%2C_families%2C_and_styles.

4.3. Estilo da fonte

Os máis empregados co seu xeito de aplicalos son:

- Negriña: `\textbf{ texto en negriña }`, `{\bf texto en negriña }` ou directamente ‘Ctrl.’+ ‘B’.
- Itálica: `\textit{ texto en itálica }`, `{\it texto en itálica }` ou directamente ‘Ctrl.’+ ‘I’.
- Subliñado: `\underline{ Texto subliñado }`.
- En caixa: `\boxed{\text{ texto en caixa }}`

4.4. Texto con cor

A maioría das cores pódense aplicar introducindo entre chaves o texto que queremos colorear coa orde `\textcolor{cor}{texto cuxa cor queremos modificar}`. Por exemplo, o código

A palabra `\textcolor{red}{lume}` aparecerá en vermello.

traducirase como

A palabra **lume** aparecerá en vermello.

L^AT_EX tamén permite definir cores, mais isto non se explicará neste documento por ser de carácter máis avanzado.

5. Listaxes

Este é outro dos moitos entornos nos que L^AT_EX supera con creces a outros sistemas de edición de textos ou editores no tocante á optimización. Unha vez aprendido o sistema, modificar a orde dos ítems dunha lista, cambiar o seu aspecto ou mesmo variar a separación entre liñas farase axiña e de xeito sinxelo.

5.1. Listaxes sen ordenar

Empregaremos a seguinte estrutura:

```
\begin{itemize}
  \itemsep0em % liña para modifizar o espazo vertical entre items.
  \item Primeiro elemento.
  \item Segundo elemento.
\end{itemize}
```

que se traducirá como

- Primeiro elemento.
- Segundo elemento.

5.2. Listaxes ordenadas

É unha estrutura moi empregada na elaboración de exercicios ou actividades nas que teñamos que listar apartados. Velaquí un exemplo:

```
\begin{enumerate}
  \itemsep0.5em % Separación de medio espazo vertical
  \item Primeiro elemento.
  \item Segundo elemento.
\end{enumerate}
```

que xerará

1. Primeiro elemento.
2. Segundo elemento.

Para modificar os indicadores dos elementos da listaxe ou, por exemplo, engadir un espazo horizontal antes ou despois de ditos indicadores deberemos engadir esta información entre corchetes² a carón do `\begin{enumerate}` do seguinte xeito:

²Precísase do paquete `enumerate` para declarar o formato dunha listaxe de maneira compacta.

```
\begin{enumerate}[\quad (i) \quad]
  \itemsep-0.2em % como vemos, admítense incluso separacións negativas.
  \item Todos os homes son mortais.
  \item Sócrates é un home.
  \item Daquela, Sócrates é mortal.
\end{enumerate}
```

que se traducirá como

- (I) Todos os homes son mortais.
- (II) Sócrates é un home.
- (III) Daquela, Sócrates é mortal.

É frecuente na elaboración de actividades para o alumnado que apareza un exercicio con varios apartados que queremos escribir de forma compacta. Para este caso, é moi recomendado incluír ao entorno `multicol` nas nosas listaxes. Por exemplo:

- | | | |
|--------------|----------------|----------------|
| a) $5x + 1$ | c) $-4 + 2y$ | e) $-5 - 7y$ |
| b) $-2x + 2$ | d) $x^2 + y^2$ | f) $-2 + 3x^2$ |

6. Imaxes

En contraste cos máis dos editores de texto convencionais nos que o usuario debe preocuparse por colocar as imaxes nos lugares máis convenientes, L^AT_EX “aforra” ese traballo ao encargarse el mesmo do anterior. As imaxes introdúcense de maneira básica co entorno `\includegraphics` (que funciona co paquete `graphicx`) atendendo á seguinte sintaxe:

```
\includegraphics[width=0.8\linewidth]{Imaxes/parabola.png}
```

Entre corchetes acláranse os atributos da imaxe e entre chaves, a ruta da mesma. No exemplo anterior, a imaxe de nome `parabola.png` terá 0.8 veces o ancho do parágrafo e o compilador importaráa dende o directorio no que se ubica, neste caso unha carpeta chamada “Imaxes”³. Aínda que existen moitas máis formas de indicar o tamaño da imaxe, de variar a súa forma, de curtala ou incluso de inclinala un ángulo determinado, usar a sintaxe do exemplo anterior considérase o máis básico e directo. En cambio, co `\includegraphics` lograremos introducir unha imaxe, pero esta reproducirase no punto exacto do texto no que a “chamemos” sen posibilidade de movela, referenciala ou poñerlle un pé de foto.

Por tanto, é aínda máis recomendable ao tratar con imaxes axudarse do entorno `\begin{figure}`, xa que nos permitirá poñerlle un pé de foto, referenciala para poder citala ao longo do documento e decidir a súa loxalización. En liñas xerais, a estrutura de código para introducir unha imaxe sería a seguinte:

³A carpeta podería ter calquer nome ou incluso o arquivo podería non estar nunha carpeta senón no propio directorio no que se ubica o `.tex`.

```

\begin{figure}[H]          % Entre corchetes indicamos a localización da imaxe.
  \centering               % Centrar a imaxe.
  \includegraphics[width=0.29\linewidth]{1_Elementos_Basicos/im/parabola.png}
  \caption{Parábola da función  $f(x)=2x^2-3x-2$ } % Pé de foto.
  \label{fig:parabola} % Referencia para poder citala.
\end{figure}

```

A continuación vemos a imaxe reproducida tal e como quedaría no noso documento:

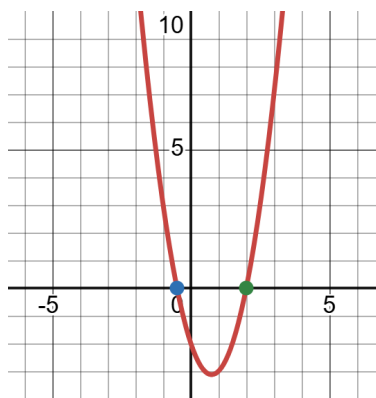


Figura 1: Parábola da función $f(x) = 2x^2 - 3x - 2$

Como xa se mencionou, L^AT_EX tratará de colocar a imaxe no lugar do documento que aporte máis “elegancia”, mellor deseño. Isto pode modificarse introducindo unha letra nos corchetes a carón do `\begin{figure}` que lle dará unha orde ao compilador, a saber:

- `t` (*top*): coloca a imaxe na parte superior da páxina (ou doutra páxina).
- `b` (*bottom*): coloca a imaxe na parte inferior da páxina (ou doutra páxina).
- `h` (*here*): coloca a imaxe, se é posible, o máis preto posible.
- `h!` (*here!*): coloca a imaxe (ínstoches a que o fagas!) o máis preto posible.
- `H` (*HERE*): coloca a imaxe obrigatoriamente aquí (funciona tendo declarado na Cabeceira `\usepackage{float}`).

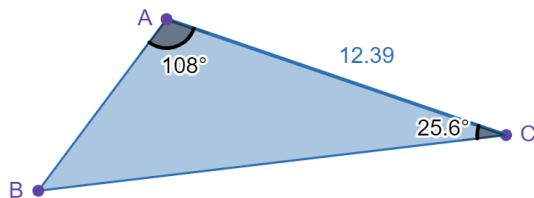
Por outra banda, referenciar a imaxe pode ser de gran utilidade se nos queremos referir a ela ao longo do documento (tanto antes como despois de que apareza). Un dos puntos fortes de L^AT_EX é precisamente as referencias cruzadas, pois están automatizadas. É dicir, non é necesario escribir manualmente a que figura nos estamos referindo. Dende o momento en que se referencia unha imaxe (ou unha táboa, unha sección...), dita referencia está “blindada” ante cambios de localización da mesma dentro do documento. É dicir, se decidimos cambiar unha imaxe de localización, L^AT_EX actualizará automaticamente o número co que etiqueta á mesma. A sintaxe para citala é:

Tal e como se observa na Imaxe `\ref{fig:parabola}`, corta ao eixo X...

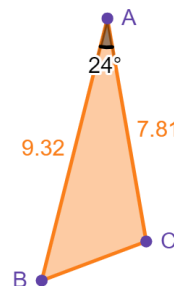
que se reproducirá como

“Tal e como se observa na Imaxe 1, corta ao eixo X...”

Tamén existe a posibilidade de introducir varias imaxes á mesma altura tratándoas coma unha soa figura co entorno `subfigure`, do paquete `subfigure`. Por exemplo, as seguintes imaxes



(a) T. obt.



(b) T. acut.

Figura 2: Triángulos a resolver no exercicio 5.

foron creadas co seguinte código:

```
\begin{figure}[H]
  \centering
  \subfigure[T. obt.]{\includegraphics[width=0.45\linewidth]{im/t1.png}}
  \hspace{1cm}
  \subfigure[T. acut.]{\includegraphics[width=0.45\linewidth]{im/t2.png}}
  \label{fig:triangulos}
  \caption{Triángulos a resolver no exercicio 5.}
\end{figure}
```

7. Táboas

Expoñer a sintaxe xeral dunha táboa, así como comentar as diversas variantes que se poderían crear, non só excedería os contidos obxecto destes apuntamentos, senón que requirirían dun documento enteiro unicamente para tratar esta temática. Por iso, a opción máis recomendable para calquer usuario que estea comezando con L^AT_EX en canto á creación de táboas é axudarse dun entorno gráfico, comunmente a web <https://www.tablesgenerator.com/>.

A esta web deberemos facilitarlle os datos que imos plasmar na táboa (pódense importar dende calquera folla de cálculo) e unicamente teremos que modificar manualmente cuestións básicas en relación á forma (bordos, estilo da fonte, aliñación dos datos dentro da columna, combinación de celdas...). Unha vez aclarado o anterior, picaremos en *Generate* e xa poderemos importar mediante *Copy to clipboard* o código da mesma ao noso .tex. Por exemplo, o código da seguinte táboa

Monomio	Coeficiente	Parte literal	Grao
$-6x^2yz^3$	-6	x^2yz^3	6
a^{13}	1	a^{13}	13
6	6	Non ten	0

Cadro 1: Clasificación de monomios.

foi xerado de maneira automática en *tables generator*:

```
\begin{table}[H]
\centering
\begin{tabular}{|c|c|c|c|} % 4 columnas con bordos verticais a ambos lados.
\hline % liña horizontal divisoria
\textbf{Monomio} & \textbf{Coeficiente} & \textbf{Parte literal} & \textbf{Grao} \\ \hline % Fin de fila.
$-6x^2yz^3$ & $-6$ & $x^2yz^3$ & $6$ \\ \hline
$a^{13}$ & $1$ & $a^{13}$ & $13$ \\ \hline
$6$ & $6$ & \text{Non ten} & $0$ \\ \hline
\end{tabular}
\caption{Clasificación de monomios.}
\label{Tab:mono}
\end{table}
```

Como é apreciable, podemos facer un paralelismo coa estrutura dunha imaxe no senso de que o entorno `\begin{tabular}` (que inclúe a táboa) fai as veces de `\includegraphics`, e o `\begin{table}` (que permite referenciala e darlle un *Caption*) fai as veces de `\begin{figure}`.

8. Creación de comandos

Se a temática tratada na sección 7 podería ser exposta nun só documento, todo o referente á creación de comandos podería perfectamente ocupar un curso completo. A creación de comandos é, probablemente, o punto máis forte e poderoso co que xoga L^AT_EX. Un bo dominio deste entorno “catapulta” ao usuario a un nivel de optimización e rentabilidade impensables de acadar con calquer outro sistema de edición de textos. A idea fundamental que subxace na creación de comandos é lograr substituír tarefas mecánicas e repetitivas por unha soa orde. Aínda que non se vai tratar o tema a un nivel intermedio, nin moito menos en profundidade⁴, si se avanzarán nesta sección algunhas pinceladas básicas. A sintaxe xeral, que deberá escribirse na Cabeceira⁵, para a creación dun novo comando é

```
\newcommand{\nomedocomando}[n]{instrución #1 instrución #2...instrución #n}
```

⁴A un nivel avanzado, o paquete `xparse` é capaz de superar certas “fronteiras” con respecto ao entorno `newcommand`, permitindo, entre outras opcións, crear comandos cun número de argumentos variable.

⁵Tamén se acepta escribilo no Corpo do documento, pero o máis recomendable é ter todos os comandos definidos no mesmo lugar (na Cabeceira predominantemente) para que sexa sinxelo atopalos se se quere modificar algo. É máis, o obxectivo é telos todos definidos no preámbulo que empreguemos habitualmente para poder facer uso deles sen ter que definilos de novo (os comandos defínense unha vez e quedan “para sempre”).

onde n é o número de variables que empregaremos para definilo e $\#1$, $\#2$, ..., $\#n$ indican as posicións dentro da definición do comando que ocuparán as variables (tamén chamadas argumentos) que introduzamos. Cando “invocemos” o comando, deberemos facelo con tantos grupos de chaves como argumentos teña o comando:

`\nomedocomando{\#1}{\#2}...\{\#n\}`

Vexamos un par de exemplos:

- I) Para aforrar tempo escribindo o valor absoluto dunha cantidade, sería moi útil crear un sinxelo comando (`\abs{}`) dunha soa variable que agrupe dentro dun valor absoluto todo o que escribamos entre chaves. Este comando podería ser creado coa seguinte sentencia: `\newcommand{\abs}[1]{\mid#1\mid}`. Unha vez declarado na Cabeceira do documento, cada vez que escribamos, por exemplo, `\abs{x^2-y^2}`, reproducirá $|x^2 - y^2|$.
- II) Imaxinemos que queremos escribir un exercicio con dez apartados para traballar a proporcionalidade co propósito de afianzar como se relacionan entre si os catro elementos dunha proporción. Necesitamos que se repita a mesma oración con cada cuaterna de números, polo que non é mala idea, en lugar de escribila dez veces, redactala unha soa vez en forma de comando para poder empregala cando o desexemos e de maneira directa:

`\newcommand{\propor}[4]{\$#1\$ é a \$#2\$ como \$#3\$ é a \$#4\$. É dicir,
 $\frac{\$#1\$}{\$#2\$} = \frac{\$#3\$}{\$#4\$}$ }`

Unha vez creado, a simple introdución de `\propor{4}{6}{12}{18}` traducirase no documento como

4 é a 6 como 12 é a 18. É dicir,

$$\frac{4}{6} = \frac{12}{18}$$

Os anteriores exemplos son unha mostra insignificante do “mundo” de tarefas que se poden optimizar con L^AT_EX. Pensemos en cantas cousas repetitivas ou mecánicas (encabezado dun boletín, puntuacións nos enunciados dun exame, enunciados de exercicios parecidos, solucións de ecuacións ou de sistemas que sempre gardan o mesmo formato...) escribimos ao elaborar materiais didácticos e que se poderían realizar cunha soa orde, facilitando a escritura do documento e, por riba de todo, permitíndonos **aforrar tempo**. Se ben é certo que crear un comando implica deter a escritura do documento propiamente dito e sacrificar uns intres para definilo, este tempo de retraso é unha inversión a longo prazo grazas á cal non teremos que volver escribir toda a tarefa en cuestión, pois xa a teremos compactada nunha soa orde. En definitiva, a filosofía da creación dun comando podería resumirse en “fame para hoxe, pan para mañá, pasadomañá, pasado-pasadomañá...”.