

Tutorial básico de L^AT_EX

Daniel Madeira

28 de julho de 2020

Tutorial básico de L^AT_EX

Um tutorial prático no formato .tex.

Sumário

Prefácio	1
Linguagem \LaTeX	2
Arquivo	2
Estrutura do código	2
Partes	2
Ambiente	3
Caracteres especiais	3
Comentário	3
Pacotes	4
Comandos	4
Definição e redefinição	4
Uso	5
Unidades de medida	6
Cores	6
Formatação da página	8
Margens	8
Quebras de página	9
Mesma página	9
Espaço vazio	9
Estilo	10
Colorindo	10
Seções no documento	11
Níveis de seção	11
Livro	12
Formatação de texto	13
Parágrafo	13
Quebras de parágrafo e linha	13
Espaçamento entre parágrafos	13
Espaçamento entre linhas	14
Espaçamento entre palavras	15
Indentação	15
Alinhamento	16
Fontes	16

Tamanho	16
Estilo	17
Família	17
Colorindo	17
Estrutura de texto	18
Tabela	18
Lista	20
Verso	22
Nota de rodapé	22
Colunas	22
Matemática	25
Em linha	25
No modo de exibição	25
Notações	27
Ligações	28
Referências	28
Ligações internas	28
Ligações web	29
Gráficos	30
Imagens externas	30
Caminho	30
Carregamento	31
Posicionamento	31
Plotagem de dados	32
Configuração	33
Ambientes	33
Plotagem	34
Exemplos de plotagem	34
Conclusão	37
Considerações	37
Onde saber mais	37
Colofão	38

Prefácio

O \LaTeX é um sistema de preparação de documento com alta qualidade para composição tipográfica. É utilizado para criar documentos dos mais variados tipos de publicação, como artigos, teses, dissertações, livros, cartas, relatórios ou qualquer outro tipo de documento. Possui um alto grau de exatidão e precisão na diagramação do conteúdo do documento e alta qualidade na formatação automática do documento. O \LaTeX é uma ampliação do original sistema de tipografia \TeX . Tornou-se um padrão para produção de documentos científicos.

O sistema \LaTeX possui código aberto e é gratuito. Está disponível para qualquer sistema operacional, produzindo o mesmo resultado em qualquer sistema. Cria arquivos pequenos e com resultados de alta qualidade. É capaz de exportar o documento para os formatos Post Script e PDF.

Este tutorial tem o propósito de mostrar o mínimo, o básico, para se conseguir produzir um documento, de uma forma prática. O próprio arquivo `.tex` deste tutorial é um exemplo básico da linguagem em \LaTeX .

Linguagem \LaTeX

Arquivo

O arquivo do código-fonte em \LaTeX deve conter apenas bytes que representem caracteres, sem nenhuma informação adicional. Trata-se do denominado arquivo de texto plano (txt). Este arquivo recebe a extensão .tex. A codificação recomendada para este arquivo é a codificação UTF-8 ou Latin1, dependendo do sistema operacional que está processando o \LaTeX .

Pode-se utilizar qualquer editor de texto puro para produzir o arquivo .tex, entretanto, é recomendável utilizar um editor \LaTeX para obter uma melhor produtividade. Basicamente, precisa estar instalado no computador: uma distribuição \TeX , por exemplo TeX Live ou MikTeX; e um editor \LaTeX , por exemplo TeXstudio, Texmaker ou TeXworks.

Atente-se: Editar um documento em \LaTeX não será WYSIWYG ("What You See Is What You Get"). Será literalmente editar o código-fonte do documento, inserindo os comandos de formatação do texto.

Estrutura do código

Partes

A estrutura global do código-fonte em \LaTeX é basicamente:

```
\documentclass{...}  
\begin{document}  
...  
\end{document}
```

A área entre `\documentclass{...}` e `\begin{document}` é denominada preâmbulo. Neste preâmbulo ficam os comandos que afetam todo o documento. Como as chamadas do uso de pacotes, definições de parâmetros de comandos, criação de novos comandos, recriação de comandos existentes etc.

A área entre `\begin{document}` e `\end{document}`, após o preâmbulo, forma o bloco principal, o ambiente do documento, é onde fica todo o conteúdo do documento.

Ambiente

Na linguagem \LaTeX , um bloco é definido entre os comandos `\begin{}` e `\end{}`. Estes comandos formam o bloco de ambiente. Os comandos inseridos dentro deste ambiente tem seu efeito restrito ao interior do bloco.

Caracteres especiais

Quase tudo pode ser digitado livremente no documento, que fará parte da impressão final, salvo alguns caracteres que são considerados especiais. Estes caracteres simbólicos são reservados pela linguagem \LaTeX porque são para introduzir comandos e possuem um significado especial: `# $ % ^ & _ { } ~ \`

Saiba a função de cada um deles:

Caractere	Função	Como imprimir no PS/PDF
<code>#</code>	parâmetro de macro	<code>\#</code>
<code>\$</code>	modo matemático	<code>\\$</code>
<code>%</code>	linha de comentário	<code>\%</code>
<code>^</code>	sobrescrito (no modo matemático)	<code>\^{}</code> or <code>\textasciicircum</code>
<code>&</code>	separador de colunas	<code>\&</code>
<code>_</code>	subscrito (no modo matemático)	<code>_</code>
<code>{ }</code>	bloco de processamento	<code>\{ \}</code>
<code>~</code>	espaço inquebrável	<code>\textasciitilde</code> or <code>\~{}</code>
<code>\</code>	início de comando	<code>\textbackslash</code> or <code>\</code>

Para usar (imprimir) algum destes caracteres no seu texto, digite com o caractere `\` ou use o comando de impressão.

Comentário

É possível inserir comentários no código-fonte do documento em \LaTeX . Os comentários de uma linha ficam após o caractere `%`. Os comentários com mais de uma linha ficam em um bloco de ambiente `{comment}`:

```
% comentário de uma linha.

\begin{comment}
    Bloco de comentário
    com mais de uma linha.
\end{comment}
```

Pacotes

O \LaTeX inclui alguns comandos básicos porém existem muitos outros comandos úteis que são implementados com o uso de pacotes no código-fonte. Alguns pacotes podem receber parâmetros:

```
\usepackage[ddmmyyyy]{datetime}
```

Comandos

O \LaTeX é uma linguagem movida por comandos (ou macros) no entorno do texto. Os comando são discriminados pelo caractere `\`, escrito em uma sintaxe como `\comando`.

O primeiro comando no código-fonte em \LaTeX é o comando `\documentclass`. Nele se define a classe do documento (ex. `article`, `report`, `book`, `letter`, `slides`, `ltnews`, `beamer` ou `memoir`) e os parâmetros para tamanho do papel, tamanho da fonte, lados de impressão etc.:

```
\documentclass[a4paper,12pt,oneside]{book}
```

Definição e redefinição

A linguagem \LaTeX permite criar novos comandos, através do comando `\newcommand`. O novo comando recebe um nome e uma definição. Isto possibilita, por exemplo, um modo de criar uma constante:

```
\newcommand{\nomeConstante}{valor}  
\newcommand{\agua}{H$_2$O}
```

Os comandos existentes podem ser redefinidos com o comando `\renewcommand`, por exemplo:

```
\renewcommand*\contentsname{Sumário} % refaz o termo para TOC.  
\renewcommand*{\thepage}{capa} % string capa no número da página
```

Criar e recriar comandos no \LaTeX vai muito além, não limitando-se somente a um valor para a definição. É possível criar combinações de comandos para compôr a definição do novo comando, inclusive a possibilidade de inserção de argumentos. A sintaxe básica é:

```
\newcommand{\nome}[n]{definição}
```


Sendo n indicando o número de argumentos para uso pelo novo comando. Por exemplo, este novo comando que define um novo modo de inserir os capítulos:

```
\newcommand{\meucapitulo}[2]{
  \setcounter{chapter}{#1}
  \setcounter{section}{0}
  \chapter*{#2}
  \addcontentsline{toc}{chapter}{#2}
}
```

Ou este, por exemplo, que cria um comando para justificar um texto com fonte teletipo (texttt):

```
\newcommand*{\justifytt}{%
  \fontdimen2\font=0.4em%
  \fontdimen3\font=0.2em%
  \fontdimen4\font=0.1em%
  \fontdimen7\font=0.1em%
  \hyphenchar\font=-%
}
```

Duas observações:

Primeiro, perceba o caractere de asterisco em `\newcommand` e `\renewcommand`. Na origem da linguagem $\text{T}_{\text{E}}\text{X}$ os comandos não podiam ter um `\par` na definição. A linguagem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ controla a possibilidade disso com a ausência ou presença do asterisco na chamada do comando. Com o asterisco, o comando não aceita parágrafos dentro da definição do comando.

Segundo, perceba o caractere `%` no fim das linhas. Alguns comandos que lidam precisamente com espaços entre os caracteres, se comportam melhor quando é inserido o `%` no final da linha.

Uso

A sintaxe geral de uso de um comando é:

```
\comando[argumento opcional]{argumento compulsório}
```

ou

```
\comando{argumento compulsório}[argumento opcional]
```

O nome do comando é sensível a letras maiúsculas e minúsculas e compõe somente de caracteres alfa-numéricos. Os argumentos podem ser mais de um, se houver.

Unidades de medida

As unidades de medidas aceitas nos comandos do LaTeX são:

pt	pontos
mm	milímetros
cm	centímetros
in	polegadas
ex	altura de um x minúsculo na fonte corrente
em	largura de um M maiúsculo na fonte corrente
mu	unidade matemática igual à 1/18em

Muitos comandos aceitam valores negativos, por exemplo `\hspace{-1.5em}`.

Cores

Com o uso do pacote `xcolor` é possível definir cores para o texto, fundo do texto, fundo da página, linhas e colunas de tabelas, gráficos etc. Pode-se usar as cores pré-definidas ou definir novas cores usando valores em RGB, Hex ou CMYK. Inicialmente, com o uso do pacote `xcolor`, existem algumas cores pré-definidas, que são:

`black, blue, brown, cyan, darkgray, gray, green, lightgray, lime, magenta, olive, orange, pink, purple, red, teal, violet, white, yellow.`

Se o pacote foi carregado com a opção `[dvipsnames]`, então um total de 68 cores estarão pré-definidas:

`Apricot, Aquamarine, Bittersweet, Black, Blue, BlueGreen, BlueViolet, BrickRed, Brown, BurntOrange, CadetBlue, CarnationPink, Cerulean, CornflowerBlue, Cyan, Dandelion, DarkOrchid, Emerald, ForestGreen, Fuchsia, Goldenrod, Gray, Green, GreenYellow, JungleGreen, Lavender, LimeGreen, Magenta, Mahogany, Maroon, Melon, MidnightBlue, Mulberry, NavyBlue, OliveGreen, Orange, OrangeRed, Orchid, Peach, Periwinkle, PineGreen, Plum, ProcessBlue, Purple, RawSienna, Red, RedOrange, RedViolet, Rhodamine, RoyalBlue, RoyalPurple, RubineRed, Salmon, SeaGreen, Sepia, SkyBlue, SpringGreen, Tan, TealBlue, Thistle, Turquoise, Violet, VioletRed, White, WildStrawberry, Yellow, YellowGreen, YellowOrange.`

Para definir novas cores, insira os comandos no preâmbulo, seguindo estes exemplos:

```
\definecolor{cinza}{gray}{0.95}
\definecolor{laranja}{RGB}{255,127,0}
\definecolor{laranja}{HTML}{FF7F00}
\definecolor{laranja}{cmyk}{0,0.5,1,0}
```

Ou ainda, crie uma mistura de cores, por exemplo:

```
\colorlet{azurelo}{blue!50!yellow}
```

Para cada lugar de uso das cores, um comando específico será utilizado. Por exemplo, o texto é colorizado com `\textcolor{cor}`, uma linha de tabela com `\rowcolor{cor}` e por aí vai. Em todos estes comandos, a cor é referenciada pelo seu nome no argumento do comando.

A cor pode ser implementada integralmente, com 100% de sua intensidade, ou reduzida em sua intensidade ou até misturada com outras cores. Para a redução de intensidade usa-se a sintaxe com seu nome + exclamação + valor, por exemplo: `blue!60`. A mistura de cores funciona acrescentando mais uma exclamação e o nome da segunda cor, que também pode ter sua intensidade reduzida, por exemplo: `blue!60!yellow`.

Um exemplo de comando completo:

```
\textcolor{red!50!violet!90}{seu texto}
```

Formatação da página

Margens

O pacote `{geometry}` proporciona um meio de configurar a disposição da página. Por exemplo, esta página está com a margem de 4,5cm e sem o espaço do cabeçalho (veja o código-fonte deste tutorial).

```
\newgeometry{margin=4.5cm,nohead}
```

Basicamente usa-se dois comandos, um para definir e outro para restaurar o que foi definido no preâmbulo:

```
\newgeometry{top=1.5cm,bottom=1.5cm,right=1cm,left=1cm}
```

```
\restoregeometry
```

Ambos os comandos implicam em uma quebra de página, para fazer valer a alteração na dimensão.

Quebras de página

Além dos comandos de geometria de página, há outros específicos para impor uma quebra na página. Basicamente são os comandos:

```
\pagebreak  
\newpage  
\clearpage
```

O comando `\pagebreak` faz com que os parágrafos se desloquem para preencher toda página, para não deixar espaço vazio no fim. Diferentemente, o comando `\newpage` não estica os espaços entre os parágrafos, deixando um grande espaço vazio no fim da página. O comando `\clearpage` é similar ao `\newpage`, apenas agindo também nas figuras (mas ainda não testei).

Mesma página

Caso tenha um conteúdo que queira manter-se em uma mesma página, sem quebra pelo meio, use o ambiente `{samepage}`.

```
\begin{samepage}  
...  
\end{samepage}
```

Espaço vazio

É possível adicionar espaços vazios entre os conteúdos na página. Basicamente existe os comandos para espaço horizontal, que empurra o próximo conteúdo horizontalmente, e para espaço vertical, que empurra o próximo conteúdo verticalmente (valores negativos realizam o inverso, contraem o espaço). Os comandos são:

```
\hspace{medida}  
\vspace{medida}
```

Exemplos:

```
\hspace{1.5em}  
\vspace{4cm}
```

Quando for lidar com letras e palavras, use a unidade de medida em, pois esta unidade é proporcional ao tamanho e família da fonte do texto.

Uma interessante utilidade, caso queira posicionar os primeiros parágrafos no topo da página e os últimos parágrafos no fim da página, use um destes dois comandos equivalentes para esticar o espaço vazio entre eles:

```
\vspace{\fill}
```

```
\vfill
```

Por exemplo, este parágrafo foi para baixo com `\vspace{3cm}`. Obs.: Usando `\vspace*` (com asterisco) o \LaTeX não remove o espaço vertical do fim da página (a documentação oficial também é confusa nesta explicação).

Estilo

Para limpar o estilo aplicado, na página atual ou nas próximas páginas, use um destes comandos com o estilo desejado:

```
\thispagestyle{empty}
```

```
\pagestyle{plain}
```

O estilo `empty` limpa tanto o cabeçalho quanto o rodapé. No estilo `plain`, que é o padrão, o cabeçalho fica vazio e o rodapé contém o número da página no centro.

Colorindo

Uma ou mais páginas podem ser coloridas com o comando `\pagecolor{}`. Este comando terá efeito da página atual em diante. Por exemplo:

```
\pagecolor{gray!10!yellow!10}
```

Para cancelar a definição da cor na página atual em diante, use o comando:

```
\nopagecolor
```

Seções no documento

Níveis de seção

O documento em L^AT_EX pode ser seccionado em até 7 níveis, dependendo da classe declarada. As divisões de conteúdo no documento podem ser:

Divisão	Nível	Comando
parte	-1	<code>\part{nome}</code>
capítulo	0	<code>\chapter{nome}</code>
seção	1	<code>\section{nome}</code>
subseção	2	<code>\subsection{nome}</code>
subsubseção	3	<code>\subsubsection{nome}</code>
parágrafo	4	<code>\paragraph{nome}</code>
subparágrafo	5	<code>\subparagraph{nome}</code>

Basta usar o comando do nível desejado e o que vier depois será desta divisão. Não há comando de encerramento, o próximo comando da próxima divisão é que indica a mudança. Exemplo:

```
\begin{document}

  \chapter{Introdução}
  ...
  \chapter{Materiais}
    \section{Líquidos}
    ...
    \section{Sólidos}
      \subsection{Descartáveis}
      ...
      \subsection{Não-descartáveis}
      ...
  \chapter{Conclusão}
  ...

\end{document}
```

Os comandos destes níveis podem ser escritos na sintaxe sem o caractere *, desta forma, são numerados, prefixados com o número e adicionados automaticamente no sumário do documento. Com a utilização do *, logo após o nome do comando, nada disso acontece, então, para incluí-los no sumário adicione mais um comando. Exemplos para algumas divisões:

```
\chapter*{Nome do Capítulo}
\addcontentsline{toc}{chapter}{Nome do Capítulo}

\section*{Nome da Seção}
\addcontentsline{toc}{section}{Nome da Seção}

\subsection*{Nome da Subseção}
\addcontentsline{toc}{subsection}{Nome da Subseção}
```

Livro

Em documentos da classe book, opcionalmente pode-se seccionar o conteúdo em quatro partes: frontal, principal, apêndice e traseira.

Tradicionalmente, a parte frontal contém a página do título, folha de rosto, resumo, sumário, prefácio, lista de figuras e lista de tabelas. A parte principal contém o conteúdo propriamente dito. Logo após existe o apêndice e a parte traseira contém o glossário, notas, bibliografia e índice.

São quatro comandos que definem estas partes:

```
\frontmatter
\mainmatter
\appendix
\backmatter
```

A parte em `\frontmatter` terá a numeração romana nas páginas e não terá os capítulos numerados. A parte em `\mainmatter` terá o comportamento padrão do documento e a sequência numérica das páginas é reiniciada. A parte `\appendix` reinicia a numeração de capítulos, usa letras na numeração de capítulos e continua seguindo a numeração das páginas principais. A parte `\backmatter` continua seguindo a numeração das páginas principais mas volta a desativar a numeração dos capítulos.

Formatação de texto

Parágrafo

Quebras de parágrafo e linha

Isto é um texto em um parágrafo. O alinhamento justificado é aplicado por padrão. A linha se estica horizontalmente para ocupar todo o espaço entre as margens.

Para iniciar um novo parágrafo basta pular uma linha no código-fonte do `LaTeX`.

Ou usar o comando `\par` no final da linha;

Para quebrá-la em um novo parágrafo.

Isto é um texto em uma linha,
o comando `\newline` ou `\\` (duas barras invertidas) faz uma quebra de linha, sem criar um novo parágrafo.

Espaçamento entre parágrafos

Os parágrafos, por padrão, não possuem um espaçamento entre eles diferente da separação simples entre linhas. Use esta combinação de comandos para definir um espaçamento dos próximos parágrafos:

```
\setlength{\parskip}{1em}
```

Perceba que este parágrafo já possui 1em de distância do parágrafo anterior e também do próximo parágrafo abaixo.

De agora em diante, todos os parágrafos terão este espaçamento entre eles.

Espaçamento entre linhas

Por padrão, ocorre o espaçamento simples entre as linhas. Alguns comandos modificam isso, do pacote `setspace`:

```
\onehalfspacing
\doublespacing
\singlespacing

\renewcommand{\baselinestretch}{0.80}\normalsize
\renewcommand{\baselinestretch}{1}\normalsize
```

Seguem os quatro exemplos para espaçamento entre as linhas de 0,80, simples(1), 1,5 e duplo:

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Espaçamento entre palavras

Além do espaço comum (proveniente da tecla de espaço) entre as palavras, existem alguns comandos que alteram o espaço para mais ou menos e também servem para forçar a colocação de espaço onde a formatação automática do \LaTeX suprime.

Comando curto e longo	Tamanho
$\backslash,$	$\backslash\text{thinspace}$ 3/18 de $\backslash\text{quad}$ (3 mu)
$\backslash:$	$\backslash\text{medspace}$ 4/18 de $\backslash\text{quad}$ (4 mu)
$\backslash;$	$\backslash\text{thickspace}$ 5/18 de $\backslash\text{quad}$ (5 mu)
$\backslash!$	$\backslash\text{negthinspace}$ -3/18 de $\backslash\text{quad}$ (-3 mu)
	$\backslash\text{negmedspace}$ -4/18 de $\backslash\text{quad}$ (-4 mu)
	$\backslash\text{negthickspace}$ -5/18 de $\backslash\text{quad}$ (-5 mu)
\backslash (espaço após a barra)	espaço normal
	$\backslash\text{quad}$ espaço da fonte corrente (18 mu)
	$\backslash\text{qqquad}$ dobro de $\backslash\text{quad}$ (36 mu)

Obs.: É comum em alguns comandos em linha, se houver uma continuação de texto após, mesmo que se tenha digitado um espaço, o conteúdo é impresso sem o espaço. Para resolver isto, inclua uma barra invertida antes deste espaço.

Indentação

Os parágrafos costumam indentar-se automaticamente. Mas, em uma instalação padrão do \LaTeX , o primeiro não se indenta, somente do segundo em diante. Neste documento, se não tivesse instalado o pacote `{indentfirst}`, este primeiro parágrafo não estaria indentado.

Para definir um tamanho de indentação, usa-se, antes dos parágrafos:

```
 $\backslash\text{setlength}\{\backslash\text{parindent}\}\{3\text{em}\}$ 
```

Por exemplo, este parágrafo está com 3em de tamanho na indentação.

Já este parágrafo não está indentado, pois antes dele há o comando `\noindent`.

Voltando à indentação normal, com `\setlength{\parindent}{1.5em}`.

O comando `\indent` força uma indentação, caso não ocorra, mas só se `\parindent` estiver diferente de zero.

Obs.: O comando `\hspace{1.5em}` colocado no início de um parágrafo simula o mesmo efeito da indentação, já o valor negativo, `\hspace{-1.5em}`, anula a indentação existente.

Alinhamento

Além do alinhamento justificado, que é o padrão, outros alinhamentos podem ser aplicados com os comandos `\begin ... \end`. Como já visto, estes comandos criam um ambiente de formatação de bloco de texto.

Conteúdo centralizado na página,
definido com `\begin{center} ... \end{center}`.

Conteúdo alinhado à esquerda,
definido com `\begin{flushleft} ... \end{flushleft}`.

Conteúdo alinhado à direita,
definido com `\begin{flushright} ... \end{flushright}`.

Fontes

Tamanho

Existem alguns tamanhos para os caracteres da fonte (e incrementado pelo pacote `{moresize}`), a partir do normal definido entre 10pt, 11pt ou 12pt, os quais são ativados por comandos com sintaxe inline ou para bloco de ambiente:

Texto miúdo	<code>\tiny{}</code>
Texto ppequeno	<code>\ssmall{}</code>
Texto script	<code>\scriptsize{}</code>
Texto rodapé	<code>\footnotesize{}</code>
Texto pequeno	<code>\small{}</code>
Texto normal	<code>\normalsize{}</code>
Texto largo	<code>\large{}</code>
Texto Largo	<code>\Large{}</code>
Texto LARGO	<code>\LARGE{}</code>
Texto imenso	<code>\huge{}</code>
Texto Imenso	<code>\Huge{}</code>
Texto IMENSO	<code>\HUGE{}</code>

A forma da sintaxe inline também pode ser assim: `{\small ... }`.

A outra maneira, de definir o tamanho do texto, é em um ambiente com bloco de parágrafo. Exemplo:

```
\begin{large}
...
\end{large}.
```

Estilo

Os diversos estilos para os caracteres são:

Texto em negrito	<code>\textbf{}</code>	<code>\bfseries</code>
Texto médio	<code>\textmd{}</code>	<code>\mdseries</code>
<i>Texto em itálico</i>	<code>\textit{}</code>	<code>\itshape</code>
<i>Texto em negrito e itálico</i>	<code>\textbf{\textit{}}</code>	
<i>Texto inclinado</i>	<code>\textsl{}</code>	<code>\slshape</code>
<u>Texto sublinhado</u>	<code>\underline{}</code>	
Texto sobrescrito	<code>\textsuperscript{}</code>	
Texto subscripto	<code>\textsubscript{}</code>	
Texto normal	<code>\text{}</code>	
Texto normal	<code>\textnormal{}</code>	<code>\normalfont</code>
Texto vertical	<code>\textup{}</code>	<code>\upshape</code>
TEXTO EM PEQUENAS MAIÚSCULAS	<code>\textsc{}</code>	<code>\scshape</code>

Família

As famílias de caracteres na fonte em uso basicamente são:

Texto romano (serifado).	<code>\textrm{}</code>	<code>\rmfamily</code>
Texto de máquina (monoespaçado).	<code>\texttt{}</code>	<code>\ttfamily</code>
Texto sem serifa.	<code>\textsf{}</code>	<code>\sffamily</code>

Mas nem todas as fontes suportam esta variação de família. Por exemplo, a fonte Arev é exclusivamente do tipo sem serifa.

Colorindo

Para colorir um texto, usa-se o comando `\textcolor{cor}{...}` ou `{\color{cor}...}`. Ou pode-se definir uma cor diretamente no comando, por exemplo:

```
\textcolor{brown!70!black}{seu texto}
```

```
{\color{brown!70!black} seu texto}
```

```
\textcolor[RGB]{190,85,0}{ seu texto }
```

```
{\color[RGB]{190,85,0} seu texto }
```

Para o fundo do texto, usa-se o comando `\colorbox{cor}{...}`:

```
\colorbox{Sepia!10}{seu texto}
```

Estrutura de texto

Tabela

Uma tabela pode ser construída com o ambiente `{tabular}`. Este comando requer um argumento que indica quantas colunas terá e qual o alinhamento de cada coluna.

Para indicar a quantidade de colunas e o respectivo alinhamento, use a quantidade de letras `l` (alinhamento à esquerda), `c` (alinhamento ao centro) e `r` (alinhamento à direita). No conteúdo, as colunas são delimitadas pelo caractere `&`. E ao final de cada linha use uma quebra.

Estrutura básica de construção de tabela:

```
\begin{tabular}{lccr}
11 & 12 & 13 & 14 \\
21 & 22 & 23 & 24 \\
31 & 32 & 33 & 34 \\
41 & 42 & 43 & 44 \\
51 & 52 & 53 & 54 \\
61 & 62 & 63 & 64 \\
\end{tabular}
```

Além do `{l c r}` no argumento, ainda há `p{largura}`, `m{largura}` e `b{largura}`, para indicar um parágrafo na coluna com alinhamento vertical no topo, meio e embaixo respectivamente. Estes argumentos definem a largura fixa da coluna.

Veja alguns exemplos de construção de tabelas (veja também pelo código-fonte). É possível traçar linhas verticais e horizontais, perfazendo um contorno. As linhas verticais são definidas com o caractere `|` entre as letras das colunas. As linhas horizontais são feitas pelo comando `\hline` ou `\cline{}`.

Uso do comando `\cline{i-f}`:

$$\begin{array}{rcccc} & & 2 & 8 & 6 \\ & \times & 8 & 2 & 6 \\ & & \hline & & 1 & 7 & 1 & 6 \\ + & & 5 & 7 & 2 & \\ & 2 & 2 & 8 & 8 & \\ & \hline & 2 & 3 & 6 & 2 & 3 & 6 \end{array}$$

Apenas com o uso do caractere |:

	Grande	Média	Pequena
Panela	5	0	3
Frigideira	2	3	3
Chaleira	2	5	1
Caçarola	7	1	0
Leiteira	4	1	3
Assadeira	4	4	0

Outro recurso é esticar uma célula por mais de uma coluna, com o comando `\multicolumn{3}{c}{conteúdo}`:

Combinações dos bonés			Saberia a cor		
Frente	Meio	Último	Frente	Meio	Último
Azul	Azul	Azul	Sim	Não	Não
Azul	Azul	Amarelo	Sim	Não	Não
Azul	Amarelo	Azul	Sim	Não	Não
Amarelo	Azul	Azul	—	Sim	Não
Azul	Amarelo	Amarelo	Sim	Não	Não
Amarelo	Azul	Amarelo	—	Sim	Não
Amarelo	Amarelo	Azul	—	—	Sim

Ou esticar uma célula por mais de uma linha, com o comando `\multirow{nlinhas}{largura}{conteúdo}`:

Seleção 70		
Goleiro	1	Félix
Defesa	4	Carlos Alberto
	2	Brito
	3	Piazza
	16	Everaldo
Meias	5	Clodoaldo
	8	Gérson
	11	Rivellino
Ataque	7	Jairzinho
	9	Tostão
	10	Pelé

Também é possível colorir as tabelas, colorindo linhas com o comando `\rowcolor{}`, colorindo colunas com o comando `\columncolor{}` e colorindo células com o comando `\cellcolor{}`:

```
\begin{tabular}{|l|l|l|l|}
\hline
\rowcolor{cyan!30}
11 & 12 & 13 & 14\\
\rowcolor{magenta!30}
21 & 22 & 23 & 24\\
\rowcolor{yellow!30}
31 & 32 & 33 & 34\\
\rowcolor{black!30}
41 & 42 & 43 & 44\\
\hline
\end{tabular}
```

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

```
\begin{tabular}{|>\columncolor{cyan!30}l
>\columncolor{magenta!30}l
>\columncolor{yellow!30}l
>\columncolor{black!30}l|}
\hline
11 & 12 & 13 & 14\\
21 & 22 & 23 & 24\\
31 & 32 & 33 & 34\\
41 & 42 & 43 & 44\\
\hline
\end{tabular}
```

11	12	13	14
21	22	23	24
31	32	33	34
41	42	43	44

Uma especificação por toda coluna pode ser aplicada no argumento do comando `{tabular}`, usando `>\comando` para comandos executados antes de cada elemento da coluna e `<\comando` para comandos que serão executados após cada elemento da coluna, como foi usado acima com o comando `\columncolor{}`.

Lista

As listas são construídas com o ambiente `{itemize}`, para listas não-ordenadas, ou com o ambiente `{enumerate}`, para listas ordenadas:

```
\begin{itemize}
\item Arroz;
\item Feijão;
\item Carne.
\end{itemize}
```

- Arroz;
- Feijão;
- Carne.

```
\begin{enumerate}
\item Arroz;
\item Feijão;
\item Carne.
\end{enumerate}
```

1. Arroz;
2. Feijão;
3. Carne.

O comando do ambiente `{itemize}` aceita especificar um caractere para indicar os itens da lista e com o ambiente `{enumerate}` também podemos definir um outro caractere de ordenação:

```
\begin{itemize}[label=\ding{71}]
```

- ✧ Arroz;
- ✧ Feijão;
- ✧ Carne.

```
\begin{enumerate}[label=\alph*.]
```

- a. Arroz;
- b. Feijão;
- c. Carne.

```
\begin{itemize}[label=$\rightarrow$]
```

- Arroz;
- Feijão;
- Carne.

```
\begin{enumerate}[label=(\roman*)]
```

- (i) Arroz;
- (ii) Feijão;
- (iii) Carne.

```
\begin{itemize}[label=\ding{43}]
```

- ☞ Arroz;
- ☞ Feijão;
- ☞ Carne.

```
\begin{enumerate}[label=\Alph*)]
```

- A) Arroz;
- B) Feijão;
- C) Carne.

Ainda é possível inserir lista dentro de item de lista, o \LaTeX altera automaticamente o símbolo de indicação de item.

Outro recurso é a lista dentro de um parágrafo, na mesma linha, por exemplo:
a) Primeiro b) Segundo c) Terceiro

Que foi produzida usando este comando:

```
\begin{inparaenum}[a)]
  \item Primeiro
  \item Segundo
  \item Terceiro
\end{inparaenum}
```

Verso

A formatação do texto para compor a estrutura de um verso é com o bloco de ambiente `{verse}`. Usa-se também a quebra de linha tradicional (`\\`) para uma simples mudança de linha, contudo, existem alguns comandos para lidar com mais eficiência em relação as quebras, por exemplo `\\>[\versewidth]`.

Duas avós com suas duas netas.
Dois maridos com suas duas esposas.
Dois pais com suas duas filhas.
Duas mães com seus dois filhos.
Duas solteiras com suas mães.
Duas irmãs com seus dois irmãos.
Leia meus dizeres mas de todo esse pessoal dito,
só falei de 6 pessoas, o que parece um mito!
Ninguém nasceu proscrito, com incesto ou com delito.
Não sei porque eu me agito e fico aflito.
Quem são eles eu peço a algum perito!

Nota de rodapé

Para inserir uma nota de rodapé¹, use o comando `\footnote{nota}` logo após a palavra que será comentada.

Colunas

Existem diversas soluções para formatar o texto em colunas. Pode-se usar os comandos de ambiente:

```
\begin{multicols} ... \end{multicols}  
\begin{vwcol} ... \end{vwcol}  
\begin{minipage} ... \end{minipage}  
\begin{tabular} ... \end{tabular}
```

Os ambientes `{multicols}` e `{vwcol}` permitem uma certa fluidez do texto entre as colunas, uma mudança automática dos parágrafos entre as colunas. Sobre o ambiente `{multicols}`, o padrão é acontecer um equilíbrio na quantidade de texto em cada lado. Este balanço automático pode ser desligado usando o ambiente `{multicols*}`.

¹Uma anotação colocada ao pé de uma página.

Veja alguns exemplos. Nos ambientes de texto fluido foi usado o comando `\columnbreak` ou o comando `\newpage` para forçar a quebra de coluna:

```
\begin{multicols}{2}  
  \lipsum[66]\par  
  \columnbreak  
  \lipsum[75]  
\end{multicols}
```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

```
\begin{minipage}[t]{0.55\linewidth}  
  \setlength{\parindent}{1.5em}  
  \lipsum[66]  
\end{minipage}\hspace{\fill}  
\begin{minipage}[t]{0.4\linewidth}  
  \setlength{\parindent}{1.5em}  
  \lipsum[75]  
\end{minipage}
```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

```

\begin{tabular}{p{0.6\linewidth}p{0.4\linewidth}}
  \parbox{0.6\textwidth}{
    \setlength{\parindent}{1.5em}
    \lipsum[66]
  }
  &
  \parbox{0.4\textwidth}{
    \setlength{\parindent}{1.5em}
    \lipsum[75]
  }
\end{tabular}

```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

```

\begin{vwcol}[widths={0.6,0.4},sep=1.5em,justify=flush,rule=0pt,indent=1.5em]
  \indent\lipsum[66]
  \newpage
  \lipsum[75]
\end{vwcol}

```

Nunc sed pede. Praesent vitae lectus. Praesent neque justo, vehicula eget, interdum id, facilisis et, nibh. Phasellus at purus et libero lacinia dictum. Fusce aliquet. Nulla eu ante placerat leo semper dictum. Mauris metus. Curabitur lobortis. Curabitur sollicitudin hendrerit nunc. Donec ultrices lacus id ipsum.

Pellentesque interdum sapien sed nulla. Proin tincidunt. Aliquam volutpat est vel massa. Sed dolor lacus, imperdiet non, ornare non, commodo eu, neque. Integer pretium semper justo. Proin risus. Nullam id quam. Nam neque. Duis vitae wisi ullamcorper diam congue ultricies. Quisque ligula. Mauris vehicula.

Matemática

O \TeX , de onde provém o \LaTeX , foi originalmente desenvolvido para facilitar a tipografia matemática, sendo capaz de formatar as mais variadas fórmulas e equações matemáticas. Assim, os recursos para este tipo de conteúdo são vastos, o grau de precisão é bastante alto.

Em linha

Para imprimir uma expressão matemática no parágrafo, na mesma linha, pode-se usar os delimitadores de ambiente $\$$ e $\$, \left($ e $\right)$ ou $\begin{\math}$ e $\end{\math}$. Por exemplo, $ax^2 + bx + c = 0$ pode ser impresso digitando $\left(ax^2 + bx + c = 0 \right)$

No modo de exibição

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Para imprimir uma expressão em uma nova linha, chamado modo de exibição, usa-se os delimitadores $\left[$ e $\right]$ ou $\begin{\displaymath}$ e $\end{\displaymath}$. Por exemplo, $\left[\frac{9}{12} + \frac{5}{34} + \frac{7}{68} = 1 \right]$ produz:

$$\frac{9}{12} + \frac{5}{34} + \frac{7}{68} = 1$$

Uma outra forma de imprimir uma expressão matemática no modo de exibição, inclusive alinhá-la, numerá-la e indexá-la no documento, é com os ambientes $\{equation\}$ ou $\{align\}$:

```
\begin{align}
(2x \times 100 + x \times 10) - (x \times 100 + 2x \times 10) &= 270 \\
200x + 10x - 100x - 20x &= 270 \\
\end{align}
```

```

90x &= 270\\
x &= 270/90\\
x &= 3
\end{align}

```

$$(2x \times 100 + x \times 10) - (x \times 100 + 2x \times 10) = 270 \quad (1)$$

$$200x + 10x - 100x - 20x = 270 \quad (2)$$

$$90x = 270 \quad (3)$$

$$x = 270/90 \quad (4)$$

$$x = 3 \quad (5)$$

```

\begin{equation*}
\begin{aligned}[c]
x + y &= 90 && \text{\footnotesize{(1ª linha multiplica por -13)}}\\
13x + 16y &= 1260 \\
\\
-13x - 13y &= -1170 \\
13x + 16y &= 1260 \\
\cline{1-2}
3y &= 90 \\
y &= 30
\end{aligned}
\begin{aligned}[c]
\mathrm{Se:} \hspace{0.5cm} x + y &= 90 \\
\\
\mathrm{então:} \hspace{0.5cm} x + 30 &= 90 \\
x &= 60
\end{aligned}
\end{equation*}

```

$x + y = 90$	(1ª linha multiplica por -13)
$13x + 16y = 1260$	
$-13x - 13y = -1170$	Se : $x + y = 90$
$13x + 16y = 1260$	ento : $x + 30 = 90$
<hr style="width: 30%; margin: 0 auto;"/> $3y = 90$ $y = 30$	$x = 60$

O caractere & marca a posição de alinhamento. Nestas sequencias de equações, foi adotada a posição no sinal de igual. O asterisco no comando cancela a numeração da equação no documento.

Notações

Veja alguns exemplos de notação de elementos matemáticos e a sintaxe dos comandos para cada um deles:

<code>\((x_1,y_2) \)</code>	(x_1, y_2)
<code>\((x^2,y^2) \)</code>	(x^2, y^2)
<code>\((x_1^2,y_2^3) \)</code>	(x_1^2, y_2^3)
<code>\(\sqrt{x} \)</code>	\sqrt{x}
<code>\(\sqrt[3]{x} \)</code>	$\sqrt[3]{x}$
<code>\(\frac{x}{y} \)</code>	$\frac{x}{y}$
<code>\(\sum_{n=1}^3 n \)</code>	$\sum_{n=1}^3 n$
<code>\(\sum\limits_{n=1}^{n=3} n \)</code>	$\sum_{n=1}^{n=3} n$
<code>\(\lim\limits_{n \rightarrow \infty} n \)</code>	$\lim_{n \rightarrow \infty} n$
<code>\(\int_0^2 x \, dx \)</code>	$\int_0^2 x \, dx$

Algumas fórmulas gigantes:

$$\sum_{j \in \mathbf{N}} b_{ij} \hat{y}_j = \sum_{j \in \mathbf{N}} b_{ij}^{(\lambda)} \hat{y}_j + (b_{ii} - \lambda_i) \hat{y}_i \hat{y}$$

$$\int_{\mathcal{D}} |\overline{\partial} u|^2 \Phi_0(z) e^{\alpha |z|^2} \geq c_4 \alpha \int_{\mathcal{D}} |u|^2 \Phi_0 e^{\alpha |z|^2} + c_5 \delta^{-2} \int_A |u|^2 \Phi_0 e^{\alpha |z|^2}$$

$$\mathbf{A} = \begin{pmatrix} \frac{\varphi \cdot X_{n,1}}{\varphi_1 \times \varepsilon_1} & (x + \varepsilon_2)^2 & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \frac{\varphi \cdot X_{n,1}}{\varphi_2 \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_2 \times \varepsilon_2} & \cdots & (x + \varepsilon_{n-1})^{n-1} & (x + \varepsilon_n)^n \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\varphi \cdot X_{n,1}}{\varphi_n \times \varepsilon_1} & \frac{\varphi \cdot X_{n,2}}{\varphi_n \times \varepsilon_2} & \cdots & \frac{\varphi \cdot X_{n,n-1}}{\varphi_n \times \varepsilon_{n-1}} & \frac{\varphi \cdot X_{n,n}}{\varphi_n \times \varepsilon_n} \end{pmatrix} + \mathbf{I}_n$$

Referências e ligações

Referências

No \LaTeX quase tudo que está numerado pode ser referenciado e o \LaTeX automaticamente atualiza as referências, se houver alguma mudança. Os objetos que podem ser referenciados são os capítulos, seções, subseções, equações, teoremas, notas de rodapé, figuras e tabelas. Os comandos para tudo isso são:

<code>\label{marcador}</code>	Usado para marcar um objeto, um identificador que será usado depois, na referência.
<code>\ref{marcador}</code>	Usado para referenciar um objeto com a respectiva marcação.
<code>\pageref{marcador}</code>	Usado para imprimir o número da página onde está o objeto com a respectiva marcação.

Há uma convenção em adotar prefixos nas marcações:

ch:	capítulo
sec:	seção
subsec:	subseção
fig:	figura
tab:	tabela
eq:	equação
lst:	lista de código
itm:	item numerado em lista

Por exemplo, aqui é uma referência à nota de rodapé do capítulo sobre formatação de texto, na página 22.

Ligações internas

Semelhante à uma página em HTML, o \LaTeX também permite *hyperlinks* dentro do documento. Podem ser ligações internas, para elementos do mesmo documento, como também ligações para arquivos externos ou endereços da Internet.

O uso do pacote `hyperref` transforma automaticamente todas as referências internas em ligações. Mas é possível adicionar trechos de textos que serão ligações para as marcações existentes, independente das referências.

Aproveitando o rótulo da marcação (`\label{marcador}`), para criar uma ligação interna, usa-se o comando `\hyperref[marcador]{texto}`.

Por exemplo, esta é uma ligação para a nota de rodapé do capítulo sobre formatação de texto.

O pacote `hyperref` possui uma extensa possibilidade de configuração. Veja o código-fonte deste tutorial para ver o comando `\hypersetup{}`, logo no preâmbulo. Este pacote também formata muitas coisas para o PDF exportado.

Ligações web

As ligações para páginas Web são criadas com os comandos `\href{}` ou `\url{}`. a diferença entre eles é que `\href` permite uma legenda para o endereço, enquanto o `\url` imprime diretamente o endereço. Segue os exemplos:

```
\href{https://www.ctan.org}{Comprehensive TeX Archive Network}
```

Comprehensive TeX Archive Network

```
\url{https://www.ctan.org}
```

<https://www.ctan.org>

Gráficos

Imagens externas

Com o uso do pacote `{graphicx}` é fácil inserir imagens no documento em \LaTeX . Além de inserir, é possível posicionar, redimensionar e rotacionar.

Caminho

Antes, pode ser útil definir previamente o caminho das imagens que irão compor o documento. O comando `\graphicspath{}` pode ser declarado no preâmbulo.

A melhor maneira é especificar o caminho relativo às imagens. Pode ser relativo ao arquivo `.tex` que carrega a imagem ou pode ser relativo ao arquivo `.tex` principal, quando há mais de um arquivo `.tex` no projeto.

Um caminho relativo ao arquivo que carrega a imagem, segue o exemplo:

```
\graphicspath{{imagens/}}
```

Um caminho relativo ao arquivo `.tex` principal, segue o exemplo:

```
\graphicspath{{../imagens/}}
```

O caminho pode ser absoluto, quando há o caminho exato ao arquivo. Os exemplos para uma situação no MS Windows e no Linux são, respectivamente:

```
\graphicspath{{c:/usuario/imagens/}}
```

```
\graphicspath{{/home/usuario/imagens/}}
```

Pode ainda combinar múltiplos caminhos no mesmo comando, se as imagens estão em mais de uma pasta. Por exemplo:

```
\graphicspath{{../imagens1/}{../imagens2/}}
```

Carregamento

O carregamento da imagem é pelo comando `\includegraphics[]{}`. Se a imagem estiver no caminho, basta informar o nome do arquivo, por exemplo:

```
\includegraphics[scale=0.2]{plataforma.jpg}
```



Figura 1: A plataforma de pesca.

O argumento opcional recebe os parâmetros para escalonamento, dimensão, rotação etc. Veja os exemplos:

```
\includegraphics[width=3cm, height=4cm]{logo.png}  
\includegraphics[width=\textwidth]{lua.jpg}  
\includegraphics[scale=1.2, angle=45]{foto.jpg}
```

Por recomendação, a extensão do arquivo pode ser omitida, assim o \LaTeX irá procurar por todos os formatos suportados de imagens.

Posicionamento

O carregamento da imagem torna-se mais preciso se estiver no ambiente `{figure}`. Com este ambiente podemos especificar o parâmetro do posicionamento:

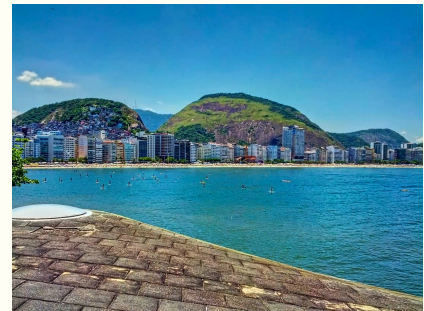
```
\begin{figure}[h!]  
  \centering  
  \includegraphics[scale=0.2]{plataforma.jpg}  
  \caption{A plataforma de pesca.}  
  \label{fig:plataforma}  
\end{figure}
```

Parâmetro	Posição
h (here)	Posição flutuante aqui mesmo.
t (top)	No topo da página.
b (bottom)	No pé da página.
p (page)	Coloca na página flutuante especial.
! (override)	Sobrepõe o cálculo do \LaTeX para a posição flutuante.

Adicionando o pacote `{wrapfig}`, o texto ganha a possibilidade de envolver a imagem carregada. Para isso usa-se o ambiente `{wrapfig}`. Segue uma ilustração:

```
\begin{wrapfigure}{r}{0.35\textwidth}
\centering
\includegraphics[width=0.35\textwidth]{copacabana.jpg}
\end{wrapfigure}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.



Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Plotagem de dados

Baseado no pacote TikZ/pgf existe o pacote `{pgfplots}`, para construir uma plotagem de dados, de funções etc. Conta com um segundo componente, o pacote `{pgfplotstable}`, que faz a formatação e o processamento de tabelas numéricas. Os ambientes de plotagem do `{pgfplots}` dependem do ambiente `{tikzpicture}`.

Provavelmente, o `{pgfplots}` é o pacote mais complexo do L^AT_EX, seu manual tem quase 600 páginas, pois possui recursos tão poderosos quanto aos disponíveis nos melhores softwares matemáticos. Vale a pena a leitura do manual.

Configuração

Antes de executar as plotagens, é interessante configurar o pgfplots e isto é feito pelo comando `\pgfplotsset`, logo no preâmbulo ou diretamente no ambiente do gráfico.

Um primeiro ponto é quanto a compatibilidade do pacote. O `{pgfplots}` é desenvolvido tendo o cuidado com as versões anteriores, com os comandos que se tornaram obsoletos ou com os comandos que a sua versão instalada ainda não suporta. Por isso, inclua o parâmetro `compat=` mais o número da versão do pacote `{pgfplots}` que está instalado em seu sistema, ou uma versão anterior, se seu documento utiliza algum comando obsoleto.

O comando `\pgfplotsset` também aceita os parâmetros de formatação do gráfico, como dimensão, estilos, fontes de caracteres etc.:

```
\pgfplotsset{width=8cm, compat=1.16}
```

Ambientes

Diversos ambientes irão estruturar uma plotagem, começando pelo ambiente gráfico `{tikzpicture}`, que forma a imagem. Interiormente, tem-se os ambientes específicos da plotagem, em relação aos eixos de escala normal ou escala logarítmica, com os ambientes `{axis}`, `{semilogxaxis}` ou `{loglogaxis}`

```
\begin{tikzpicture}
  \begin{axis}[]
    ...
  \end{axis}
\end{tikzpicture}
```

Os ambientes dos eixos podem receber um argumento com os parâmetros para formatar cada eixo, separados por vírgulas e seguindo a boa prática de digitá-los um por linha, por exemplo:

```
\begin{axis}[
  title = Título,
  xlabel = {$x$},
  ylabel = {$y$},
]
```

Dica: Externamente ao ambiente `{tikzpicture}` pode-se colocar o ambiente `{figure}`, que permite, por exemplo, adicionar uma legenda à figura e outros tratamentos.

Plotagem

Dentro do ambiente do eixo insere-se o comando para adicionar uma plotagem, com `\addplot` ou `\addplot3` para visualização em 3D. Neste comando, especifica-se a origem dos dados, se será por uma função, por coordenadas ou fornecidos por uma tabela em um arquivo externo. Também pode receber um argumento com diversos parâmetros:

```
\addplot[
    blue,
    domain=-6:4,
]
{x^2 + 2*x + 1};

\addplot[
    red,
    mark=square,
]
coordinates {(1,35)(2,34)(3,30)(4,26)(5,20)(6,17)};

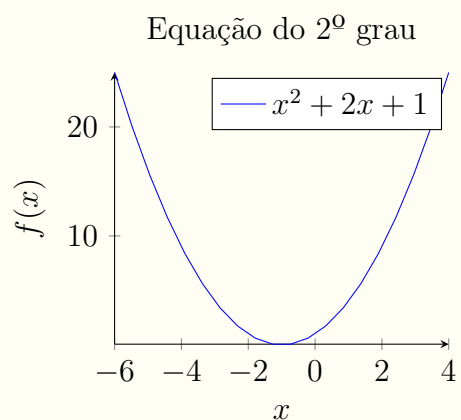
\addplot table {dados.txt};
```

O comando `\addplot` pode ser usado mais de uma vez no mesmo gráfico, caso tenha outros dados compatíveis ao mesmo domínio.

Exemplos de plotagem

Função matemática:

```
\begin{tikzpicture}
  \begin{axis}[
    axis lines = left,
    title = {Equação do 2º grau},
    xlabel = {$x$},
    ylabel = {$f(x)$},
  ]
    \addplot[domain = -6:4,
      samples = 20,
      color = blue,]
      {x^2 + 2*x + 1};
    \addlegendentry{$x^2 + 2x + 1$}
  \end{axis}
\end{tikzpicture}
```



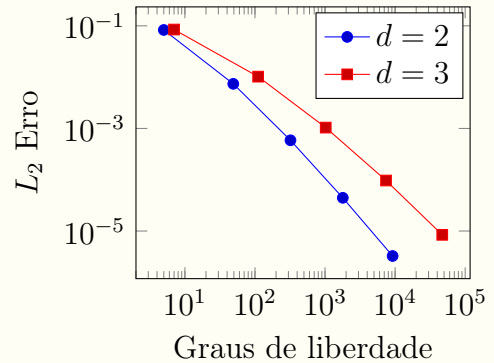
Coordenadas logarítmicas:

```
\begin{tikzpicture}
  \begin{loglogaxis}[
    xlabel = {Graus de liberdade},
    ylabel = {$L_2$ Erro},]

    \addplot coordinates {
      (5,8.312e-02) (49,7.407e-03)
      (321,5.874e-04) (1793,4.442e-05)
      (9217,3.261e-06) };

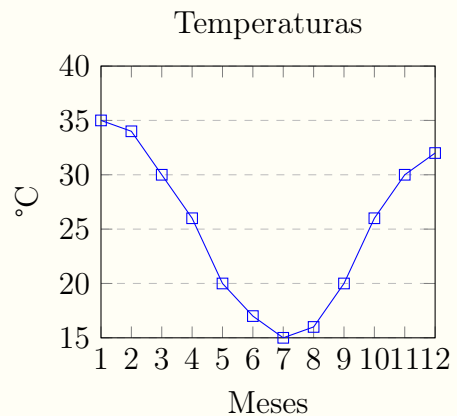
    \addplot coordinates {
      (7,8.472e-02) (111,1.022e-02)
      (1023,1.039e-03) (7423,9.658e-05)
      (47103,8.437e-06) };

    \legend{$d=2$,$d=3$}
  \end{loglogaxis}
\end{tikzpicture}
```



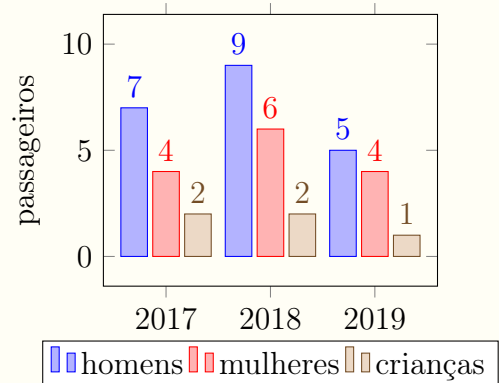
Coordenadas em escala normal:

```
\begin{tikzpicture}
  \begin{axis}[
    title = {Temperaturas},
    xlabel = {Meses},
    ylabel = {°C},
    xmin = 1, xmax = 12,
    ymin = 15, ymax = 40,
    xtick = {1,2,3,4,5,6,7,8,9,10,11,12},
    ytick = {0,15,20,25,30,35,40},
    ymajorgrids = true,
    grid style = dashed,]
    \addplot[blue,
      mark = square]
      coordinates {
        (1,35) (2,34) (3,30) (4,26) (5,20) (6,17)
        (7,15) (8,16) (9,20) (10,26) (11,30) (12,32)};
  \end{axis}
\end{tikzpicture}
```



Plotagem em barra:

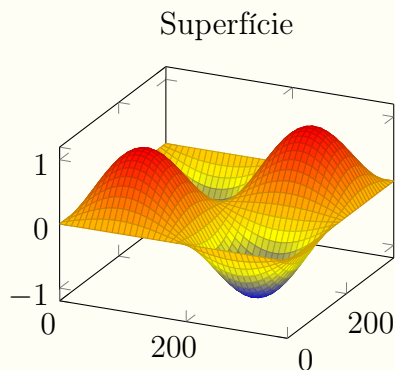
```
\begin{tikzpicture}
\begin{axis}[
  ybar,
  enlargelimits = 0.3,
  legend style = {at={(0.5,-0.2)},
    anchor = north,legend columns = -1},
  ylabel = {passageiros},
  symbolic x coords={2017,2018,2019},
  xtick = data,
  nodes near coords,
  nodes near coords align={vertical},]
\addplot coordinates {(2017,7) (2018,9) (2019,5)};
\addplot coordinates {(2017,4) (2018,6) (2019,4)};
\addplot coordinates {(2017,2) (2018,2) (2019,1)};
\legend{homens,mulheres,crianças}
\end{axis}
\end{tikzpicture}
```



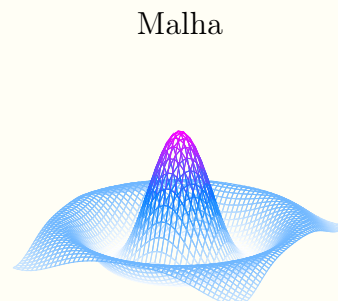
Plotagem 3D:

A plotagem em 3 dimensões é com o comando `\addplot3`. Veja dois interessantes exemplos com os parâmetros `surf` e `mesh`, que criam uma superfície e uma malha, respectivamente:

```
\begin{tikzpicture}
\begin{axis}[title = {Superfície},]
\addplot3[surf, domain = 0:360,
  samples = 40,]
{sin(x)*sin(y)};
\end{axis}
\end{tikzpicture}
```



```
\begin{tikzpicture}
\begin{axis}[title = Malha, hide axis,
  colormap/cool,]
\addplot3[mesh, samples = 50,
  domain = -8:8,]
{sin(deg(sqrt(x^2+y^2)))/sqrt(x^2+y^2)};
\end{axis}
\end{tikzpicture}
```



Conclusão

Considerações

O \LaTeX possui zilhões de comandos e quase sempre há mais de uma solução possível para formatar um conteúdo. Entretanto, o \LaTeX padrão não possui todos os comandos instalados, trazendo apenas o básico, sendo assim, existe a necessidade desta adição de pacotes. Os pacotes implementam novos comandos ao \LaTeX .

Coloquei alguns pacotes neste arquivo e nem todos estão sendo usados nos comandos. Estão aqui somente para você saber que eles existem, pois são relativamente famosos. Porém, muitos exemplos de tipografia não estão aqui, o \LaTeX é capaz de muito mais!

Quanto aos pacotes, é muito comum ter pacotes que são aprimoramentos ou reescrita de outros. Alguns pacotes pedem até uma certa ordem no carregamento. O manual de cada um deles esclarece estes detalhes. Consulte o manual de cada pacote para conhecer mais possibilidades de formatação.

Onde saber mais

- $\text{\LaTeX} 2_{\epsilon}$ The macro package for \TeX
- Comprehensive TeX Archive Network (CTAN)
- TeX Users Group (TUG)
- The Not So Short Introduction to $\text{\LaTeX} 2_{\epsilon}$
- Formatting Information - An introduction to typesetting with \LaTeX
- $\text{\LaTeX} 2_{\epsilon}$: An unofficial reference manual
- Getting Started with \LaTeX
- Uma não tão pequena introdução ao $\text{\LaTeX} 2_{\epsilon}$
- Wikibooks \LaTeX
- The Comprehensive \LaTeX Symbol List

Colofão

Este eBook foi desenvolvido usando o sistema de preparação de documento L^AT_EX 2_ε e editado com o software TeXstudio no sistema operacional Linux Fedora. O documento foi convertido para o formato PDF pelo TeX Live com o pdfTeX. O corpo do texto utiliza a fonte Computer Modern, no tamanho 12pt e as páginas possuem o tamanho A4, com três centímetros nas margens superior e esquerda, e com 2,5 centímetros nas margens direita e inferior.

As imagens da capa e da contracapa foram obtidas pelo website Pexels.com. A fotografia da capa foi obtida no endereço [https:// www.pexels.com/ photo/ photo-of-clouds-during-daytime-2088205/](https://www.pexels.com/photo/photo-of-clouds-during-daytime-2088205/) e a fotografia da contracapa foi obtida no endereço [https:// www.pexels.com/ photo/ photo-of-airplane-with-smoke-trail-2088203/](https://www.pexels.com/photo/photo-of-airplane-with-smoke-trail-2088203/). Ambas estão creditadas a Eberhard Grossgasteiger.

O código-fonte deste Tutorial foi compilado em 28 de julho de 2020.

