

Intro LaTeX

Rodrigo Girão Serrão
rodrigogiraoserrao@gmail.com

December 7, 2020

Contents

1	Introdução	2
1.1	Motivação	2
1.2	Propósito deste documento	2
2	Workshop	3
2.1	Conteúdo para iniciantes	3
2.2	Conteúdo intermédio	15
2.3	Conteúdo avançado	20
A	Exemplo de secção no apêndice	21

1 Introdução

1.1 Motivação

O workshop “Intro LaTeX” foi criado com o objetivo de introduzir os alunos da Licenciatura em Matemática Aplicada e Computação ao sistema LaTeX. LaTeX é uma ferramenta útil para estes alunos porque, por um lado, há vários trabalhos no curso que requerem a redação de um relatório e, por outro, porque esses relatórios estão invariavelmente populados por várias fórmulas matemáticas e outros objetos complexos. É verdade que, com o passar do tempo, se tem tornado *menos difícil* de usar programas como o Microsoft Word para redigir documentos que contenham fórmulas matemáticas e que sejam elegantes. No entanto, LaTeX continua a ser a opção mais viável, de longe.

1.2 Propósito deste documento

Este documento tem dois propósitos:

1. Primeiramente, serve como *template* para os participantes do workshop e para todos os que queiram um documento simples com muitas funcionalidades demonstradas.
2. Em segundo lugar, serve para me ajudar a “mim” (ou a qualquer outro palestrante) a dar o workshop de LaTeX.

2 Workshop

2.1 Conteúdo para iniciantes

Criar um ficheiro .tex

Preencher o cabeçalho com `\documentclass{article}` Falar das configurações, e.g. mudar o tamanho do texto com `\documentclass[11pt]{article}`.

Explicar que o conteúdo do documento está entre `\begin{document}` e `\end{document}`

Explicar o que são ambientes Secções do documento em que o contexto é diferente e, por esse motivo, certas coisas são interpretadas de forma especial.

Escrever uma frase e compilar Explicar que a compilação é o processo através do qual tudo o que escrevemos é visto, interpretado e transformado num pdf (ou no que quer que seja).

Estilizar o texto com `textbf`, `textit` e `texttt` Os comandos começam com `text` e as duas letras seguintes indicam o tipo de formatação:

- `bf` para “*boldface*”, i.e. para **escrever texto em bold**.
- `it` para “*italics*”, i.e. para *escrever texto em itálico*.
- `tt` para “*teletypewriter*”, i.e. para texto num tipo de letra com monoespaçamento.

Explicar que os comandos aceitam argumentos dentro de `{}` Exemplo, `\textbf{texto a bold}` produz **texto a bold**.

Falar do que são *packages* Comandos que se põem no preâmbulo para adicionar funcionalidades. (Semelhante a um `import *` em Python.)

Incluir a *package* `inputenc` para lidar com caracteres latinos, e.g. ãáà, com a opção `utf8`

Falar de como mudar de linha

- uma quebra de linha é para escrever linhas mais curtas;
- duas quebras de linha consecutivas mudam o texto de linha;
- escrever `\\` também muda de linha;
- usar `\newline` para mudar de linha;
- usar `\vspace{length}` para espaço vertical arbitrário.

Por exemplo, esta frase foi escrita em duas linhas diferentes do documento físico.

Por outro lado, esta tem uma linha branca vazia a separá-la da linha de cima. Também temos que `\\` faz com que o texto mude de linha.

Para concluir, para conseguir que este texto ficasse tão afastado do resto

da frase, tivémos de usar o comando `\vspace{3em}`.

Explicar que `$` e `$$` são usados para “escrever matemática” Explicar que `$` é usado para números/fórmulas que ficam corridos com o texto, por exemplo para mostrar que “x mais dois” fica $x + 2$, e dizer que `$$` é usado para rodear expressões a que queremos dar destaque, tais como

$$x + 3 .$$

Mencionar o site detexify.kirelabs.org Site onde desenhamos os símbolos e eles dão-nos o comando.

Package `amsfonts` para os comandos `mathbb` e `mathcal` `\mathbb{R}` é usado para produzir o símbolo dos números reais \mathbb{R} e `\mathcal{L}` produz letras caligrafadas: \mathcal{L} .

Package `amsmath` para comandos como `implies` `\implies` é usado para \implies .

Mostrar uma série de símbolos comuns Olhar para a tabela que se segue.

Table 1: Alguns símbolos e comandos respetivos (muitos precisam de `amsfonts` ou `amsmath`)

Símb.	Comando	Símb.	Comando
\mathbb{C}	<code>\mathbb{C}</code>	\mathbb{R}	<code>\mathbb{R}</code>
\mathbb{Q}	<code>\mathbb{Q}</code>	\mathbb{Z}	<code>\mathbb{Z}</code>
\mathbb{N}	<code>\mathbb{N}</code>	\implies	<code>\implies</code>
\iff	<code>\iff</code>	\equiv	<code>\equiv</code>
\approx	<code>\approx</code>	\neq	<code>\neq</code>
\lim	<code>\lim</code>	\sum	<code>\sum</code>
\prod	<code>\prod</code>	\int	<code>\int</code>
α	<code>\alpha</code>	π	<code>\pi</code>
ϕ	<code>\phi</code>	Φ	<code>\Phi</code>
\rightarrow	<code>\rightarrow</code>	∞	<code>\infty</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>

Comando `\sqrt` para raízes quadradas Usar `\sqrt{3\pi - \infty}` dá $\sqrt{3\pi - \infty}$.

Falar do ambiente `equation` Centra as equações e numera-as automaticamente. Por exemplo, “x mais dois”

$$x + 2 \tag{1}$$

Explicar como impedir numeração em geral com `*` Por exemplo, `\begin{equation*} ... \end{equation*}` cria uma equação sem número:

$$x + 3$$

Ensinar a usar `^` para potências e similares, e `_` para índices e similares Mostrar x^2 e v_i .

Ensinar `\{` para agrupar várias coisas agarradas a `^` ou `_` Por exemplo, `\lim_{x \rightarrow \infty} f(x)` cria

$$\lim_{x \rightarrow \infty} f(x)$$

e `\int_0^{2\pi + \pi^2} f(x)dx` cria

$$\int_0^{2\pi + \pi^2} f(x) dx$$

Ensinar frações `\frac{num}{den}` produz $\frac{num}{den}$.

Alinhar equações com o ambiente align Usamos `&` para indicar o ponto de alinhamento e usamos `\\` para dizer que mudamos de linha. Por exemplo,

$$x = 1 + \pi \tag{2}$$

$$y = 0 \tag{3}$$

foi produzido com

```
\begin{align}
x &= 1 + \pi \\
y &= 0
\end{align}
```

Agrupar equações com o sub-ambiente cases Usamos `\\` para dizer que mudamos de linha. Por exemplo,

$$f(x) = \begin{cases} 0, x < 0 \\ 1, x \geq 0 \end{cases} \tag{4}$$

foi produzido com

```
\begin{equation}
f(x) = \begin{cases}
0, x < 0 \\
1, x \geq 0
\end{cases}
\end{equation}
```

Obrigar o LaTeX a pôr espaços onde queremos com ~ Por exemplo, para pôr um espaço na equação em cima entre a vírgula e as condições:

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{5}$$

foi produzido com

```

\begin{equation}
f(x) = \begin{cases}
0, & \sim x < 0 \\
1, & \sim x \geq 0
\end{cases}
\end{equation}

```

Escrever texto dentro de ambientes matemáticos com text Por exemplo, a segunda condição em cima podia ser

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & \text{c.c.} \end{cases} \quad (6)$$

que foi produzido com

```

\begin{equation}
f(x) = \begin{cases}
0, & \sim x < 0 \\
1, & \sim \text{c.c.}
\end{cases}
\end{equation}

```

Ter cuidado com o estilo com que se escreve LaTeX Escrever à badalhoco é meio caminho andado para se escrever um documento impossível de editar posteriormente.

Funções trigonométricas et al, com os próprios nomes Certas funções e/ou símbolos existem como comandos com o próprio nome, por exemplo `\sin(\pi)`, `\min(0, 1)` e `\gcd(1, 2)` para $\sin(\pi)$, $\min(0, 1)$ e $\gcd(1, 2)$

Tamanho vertical de parêntesis com left e right Se expressões dentro de parêntesis (ou entre dois delimitadores, em geral) ficarem muito altas, podemos usar `\left` e `\right` para acertar o tamanho vertical. Compare-se, por exemplo,

$$\sin\left(\sum_0^\infty x_n\right) \text{ vs } \sin\left(\sum_0^\infty x_n\right)$$

que foi escrito com

```

\sin(\sum_0^\infty x_n) ~ \text{vs} ~
\sin\left( \sum_0^\infty x_n \right)

```

Usar etiquetas (label) para nos referirmos a outros objetos Todos os objetos que são numerados podem ser referenciados. Para tal, escolher um nome que identifique *claramente* esse objeto e pôr esse nome dentro do comando `\label`. Mais tarde, fazer referência ao objeto através do comando `\ref` que aceita o nome da etiqueta como argumento.

Por exemplo, o código

```
\begin{equation}
\label{eq:quadratic_formula}
x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}
\end{equation}
```

produz a equação:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (7)$$

e agora podemos falar especificamente da equação 7, através da utilização do comando `\ref{eq:quadratic_formula}`. Para fórmulas, também fica bem usar o comando `\eqref{eq:quadratic_formula}`, que tem um estilo ligeiramente diferente, para podermos falar da equação (7).

Usar prefixos nas etiquetas para agrupar objetos da mesma família

Por exemplo, `eq:` para equações, `fig:` ou `img:` para imagens, `t:` ou `tab:` para tabelas, `sec:` para secções e `subsec:` para subsecções, etc. Ser **consistente**.

Usar section, subsection e subsubsection Estes comandos partem o documento em secções, subsecções e sub-subsecções. A numeração é automática. O único argumento é o nome da divisão. Por exemplo, o código

```
\section{Introdução}
\subsection{Motivação}
\subsection{Propósito deste documento}
\section{Workshop}
\subsection{Conteúdo para iniciantes}
\subsection{Conteúdo intermédio}
\subsection{Conteúdo avançado}
```

foi usado para separar este documento nas partes que se encontram no índice.

Criar um índice com tableofcontents Basta usar o comando `\tableofcontents` no sítio em que queremos o índice.

Criar uma capa com maketitle Especificar informações como `author` e `title` no preâmbulo, e.g. com

```
\title{Intro LaTeX}
\author{Rodrigo Girão Serrão \\\
        rodrigogiraoserrao@gmail.com
}
```

e depois incluir a capa com `\maketitle`

Obrigar o resto da página a ficar em branco com clearpage Por exemplo, quando queremos que o índice não seja logo seguido de texto, usamos `\clearpage`, por isso é que há tanto espaço em branco na primeira página deste documento, porque o documento começa (mais ou menos) com

```
\begin{document}
\maketitle
\tableofcontents
\clearpage
\section{Introdução}
\subsection{Motivação}
...
```

Incluir imagens com graphicx O pacote `graphicx` dá-nos acesso a um ambiente `figure` que usamos para incluir imagens. Se tivermos uma imagem no caminho `graphics/image.png`, podemos incluí-la, centrá-la e mudar o seu tamanho com

```
\begin{figure}
\centering
\includegraphics[scale=0.3]{graphics/crowd.jpg}
\caption{Legenda da imagem}
\label{img:first-example}
\end{figure}
```

que produz a imagem 1.

A etiqueta tem de vir depois da legenda É a legenda que atribui o número à imagem, logo convém haver um número para lhe podermos atribuir uma etiqueta.

A opção scale rescala a imagem de forma proporcionada



Figura 1: Legenda da imagem

Alternativamente, a opção `width` com `\linewidth` especifica o comprimento

Usar o pacote `float` para obrigar a imagem a ficar no lugar Abrindo o ambiente com `\begin{figure}[H]` obriga a imagem a ficar no sítio onde foi definida. Por exemplo, a imagem está aqui:



Figura 2: Legenda da imagem

graças ao código (depois de pôr `\usepackage{float}` no preâmbulo)

```
\begin{figure}[H]
  \centering
  \includegraphics[width=0.5\linewidth]{graphics/crowd.jpg}
  \caption{Legenda da imagem}
  \label{img:second-example}
\end{figure}
```

Bibliografia básica Usar o ambiente `\begin{thebibliography}{99} ... \end{thebibliography}`. Dentro do

ambiente, cada linha que comece com `\bibitem{lbl}` é uma entrada. `lbl` é um nome de etiqueta para citarmos a bibliografia. Por exemplo, a bibliografia

Referências

[1] Wikipedia, lol

[2] Outro site qualquer

foi inserida com

```
\begin{thebibliography}[99]
  \bibitem{wiki} Wikipedia, lol
  \bibitem{site} Outro site qualquer
\end{thebibliography}
```

(e com uma batota no preâmbulo para ter o nome em português).

Citar a bibliografia com cite Eu sei isto, cf. [1]! (`\cite{wiki}` para a citação).

Links no próprio pdf com hyperref Usar o pacote `hyperref` faz com que citações e referências sejam clicáveis (bem como o índice) e permitem-nos ir diretamente para o objeto referido.

Links externos com href O comando `\href{url}{text}` cria links externos, e.g. para a internet, com texto personalizado e `\url{url}` cria links, e.g. este link vai para o Google, e este também: `https://google.com`. `\href{https://google.com}{este link vai para o Google}` vs. `\url{https://google.com}`

Listas Usar o ambiente `itemize` e comando `item` para fazer uma lista não ordenada. A lista

- Item 1
- Item 2
- Item 3

cria-se com o código

```
\begin{itemize}
  \item Item 1
  \item Item 2
  \item Item 3
\end{itemize}
```

Listas dentro de listas Usar um ambiente `itemize` dentro de um `item` cria outra lista. De facto, a lista

- Item 1
 - Sub 1
 - Sub 2
- Item 2
- Item 3
 - Outro Sub 1
 - Outro Sub 2

é criada com o código

```
\begin{itemize}
  \item Item 1 \begin{itemize}
    \item Sub 1
    \item Sub 2
  \end{itemize}
  \item Item 2
  \item Item 3 \begin{itemize}
    \item Outro Sub 1
    \item Outro Sub 2
  \end{itemize}
\end{itemize}
```

Enumerações O ambiente `enumerate` é semelhante ao `itemize`, mas serve para listas ordenadas. Se pegarmos no código em cima e trocarmos todos os `itemize` por `enumerate`, obtemos o seguinte:

1. Item 1

- (a) Sub 1
 - (b) Sub 2
2. Item 2
3. Item 3
- (a) Outro Sub 1
 - (b) Outro Sub 2

Tabelas Ambiente `table`. Tal como com as imagens, podes usar `centering` e `[H]` para centrar e pôr a tabela onde a escreveste. Dentro do `table`, ambiente `tabular` é usado para escrever os dados, separados por `&` e o argumento do `tabular` é uma sequência de letras `lcr` para indicar se cada coluna é alinhada à esquerda, ao centro ou à direita, respetivamente. (Há tantas letras quanto colunas.) O código

```
\begin{table}[H]
  \centering
  \begin{tabular}{lccr}
    um & dois & três &  $3 + 3 + \pi$  \\
    1 & 2 & 3 & três mais três mais pi
  \end{tabular}
  \caption{Tabela básica}
\end{table}
```

produz a tabela

um	dois	três	$3 + 3 + \pi$
1	2	3	três mais três mais pi

Table 2: Tabela básica

Usa | cenas para linhas verticais na tabela O código

```
\begin{table}[H]
  \centering
  \begin{tabular}{|l|cc|r}
    um & dois & três &  $3 + 3 + \pi$  \\
    1 & 2 & 3 & três mais três mais pi
  \end{tabular}
  \caption{Tabela básica}
\end{table}
```

produz a tabela

um	dois	três	$3 + 3 + \pi$
1	2	3	três mais três mais pi

Table 3: Tabela básica

Usa `\hline` para linhas horizontais na tabela O código

```
\begin{table}[H]
\centering
\begin{tabular}{|l|cc|r}
\hline
um & dois & três &  $3 + 3 + \pi$  \\
\hline
1 & 2 & 3 & três mais três mais pi \\
\hline
\end{tabular}
\caption{Tabela básica}
\end{table}
```

produz a tabela

um	dois	três	$3 + 3 + \pi$
1	2	3	três mais três mais pi

Table 4: Tabela básica

Usa <https://tablesgenerator.com> para gerar tabelas mais facilmente

Cria matrizes com ambientes como `bmatrix` e `pmatrix` O ambiente `exato` altera o aspeto dos parêntesis à volta da matriz. Separamos os elementos numa linha com `&` e separamos linhas com `\\`. Exemplo:

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Produzido com o código

```


$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$


```

As linhas da matriz podem, e devem, estar em linhas separadas
 Por exemplo, a matriz

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ \pi & 2\pi & 3\pi & 4\pi \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

(não é invertível e) foi produzida com o código

```


$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ \pi & 2\pi & 3\pi & 4\pi \\ 5 & 6 & 7 & 8 \end{bmatrix}$$


```

Cria matrizes online em <https://codecogs.com/latex/eqneditor.php>
 Ou edita qualquer tipo de equação lá.

Cria links para equações em <https://mathspp.com/texpaste> Para partilhares equações diretamente em qualquer meio, para que o utilizador não tenha de tentar decifrar o código LaTeX. Por exemplo, este link mostra a solução de uma equação quadrática.

2.2 Conteúdo intermédio

A partir daqui estamos a aprender funcionalidades que um utilizador intermédio de LaTeX usaria.

Incluir código com o pacote listings Usar o ambiente `lstlisting`.
 Por exemplo:

```

if __name__ == "__main__":
    print("Hello , World!")

```

é gerado com

```

\begin{lstlisting}[language=Python]
if __name__ == "__main__":

```

```

        print("Hello, World!")
\end{lstlisting}

```

Incluir código diretamente de um ficheiro com `lstinputlisting`

Por exemplo, se o ficheiro `pythoncode.py` estiver no mesmo diretório que este ficheiro `.tex`, então o comando

`\lstinputlisting[language=Python]{pythoncode.py}` produz:

```

if __name__ == "__main__":
    print("Hello , World!")

```

Pesquisar online por estilos pré-definidos para usar com código

Por exemplo, no preâmbulo deste documento defini um estilo `mystyle` e agora o comando

`\lstinputlisting[language=Python,style=mystyle]{pythoncode.py}` produz

```

1 if __name__ == "__main__":
2     print("Hello, World!")

```

Continua feio, mas está *muito* melhor.

Comando `lstinline` para código em texto corrido

A utilização do comando

```

\lstinline[language=Python,style=mystyle]
{if __name__ == "__main__": print("Hey")}

```

permite-me escrever `if __name__ == "__main__": print("Hey")`.

Alinhamento múltiplo com `alignat` Para produzir uma equação com vários alinhamentos (2 ou mais) como a seguinte, usa-se o ambiente `alignat`, cujo argumento é o número de alinhamentos verticais. O primeiro alinhamento de cada linha faz-se com `&` e os seguintes com `&&`.

$$\mathcal{L}u = f, \quad \text{in } \Omega \quad (8)$$

$$u = g + g^2, \text{ on } \partial\Omega \quad (9)$$

produzido com

```

\begin{alignat}{2}
\mathcal{L}u &= f, && \text{in } \Omega \\
u &= g + g^2, && \text{on } \partial\Omega
\end{alignat}

```


Alinhamento múltiplo dentro de outros ambientes com aligned A equação

$$u \text{ solves } \begin{cases} \mathcal{L}u = f, & \text{in } \Omega \\ u = g + g^2, & \text{on } \partial\Omega \end{cases}$$

foi produzida com o código

```


$$u \sim \text{\texttt{\textbackslash text{solves}}} \sim$$


$$\begin{cases}$$


$$\begin{aligned}$$


$$\mathcal{L}u &= f, \sim \&\&\text{\texttt{\textbackslash text{in}}} \sim \Omega \backslash \backslash$$


$$u &= g + g^2, \sim \&\&\text{\texttt{\textbackslash text{on}}} \sim \partial\Omega$$


$$\end{aligned}$$


$$\end{cases}$$


```

Figuras lado a lado com subcaption Usar o pacote `subcaption` dá acesso ao ambiente `subfigure`, que permite pôr imagens lado a lado. O ambiente aceita um argumento que é o espaço horizontal a usar.



(a) Sub 1 tem imenso texto aqui que obviamente não cabe numa só linha



(b) Sub 2

Figura 3: Legenda principal

Podemos falar da figura toda com 5 ou das sub-imagens com 5a e 5b. O código para pôr as imagens é

```


$$\begin{figure}[H]$$


$$\begin{center}$$


$$\begin{subfigure}{0.49\textwidth}$$


$$\begin{center}$$


$$\includegraphics[width=0.99\textwidth]{graphics/crowd.jpg}$$


$$\caption{\end{center}$$


```

```

        Sub 1 tem imenso texto aqui
        que obviamente não cabe numa só linha
    }
    \label{fig:sub1}
\end{subfigure}
\begin{subfigure}{0.49\textwidth}
    \centering
    \includegraphics[height=4em]{graphics/crowd.jpg}
    \caption{Sub 2}
    \label{fig:sub2}
\end{subfigure}
\caption{Legenda principal}
\label{fig:big-figure}
\end{figure}

```

e para as referir é `\ref{fig:big-figure}` ou `\ref{fig:sub2}`, por exemplo.

Alinhar as sub-legendas com [b] ou [t] Opções para o ambiente `subfigure`. Alinhadas pelo *top* ou *bottom* das legendas. Efeito (imagens mais pequenas para ocuparem menos espaço):



(a) Sub 1 tem imenso texto aqui que obviamente não cabe numa só linha



(b) Sub 2

Figura 4: `\begin{subfigure}[t] ...`

e



(a) Sub 1 tem imenso texto aqui que obviamente não cabe numa só linha



(b) Sub 2

Figura 5: `\begin{subfigure}[b] ...`

Criar os próprios comandos com newcommand Definem-se no preâmbulo e a sintaxe é `\newcommand{\cmdname}[nargs]{syntax}` onde `nargs` é o número de argumentos que o comando aceita, e para nos referirmos a esses argumentos usamos `#1`, `#2`, etc.

Por exemplo, inserir `\newcommand{\partder}[2]{\frac{\partial #1}{\partial #2}}` no preâmbulo permite que o código `$\partder{f}{x}$` gere $\frac{\partial f}{\partial x}$.

Criar um apêndice com appendix Se usarmos o comando `\appendix`, a partir daí todas as secções, subsecções, etc, ficam identificadas como pertencendo ao apêndice e numeradas de forma diferente. Vejam, por exemplo, a secção A.

**Ignorar caracteres especiais com ** Há vários caracteres que são especiais. Usamos `\` para os ignorar, por exemplo `\$` produz `$`.

Mudar o nome das figuras, tabelas, apêndice, etc. Há que descobrir o comando `name` que nos interessa e alterá-lo com `\renewcommand{\namecmd}{Texto novo}`. Por exemplo, `\renewcommand{\figurename}{Figura}` foi o que usei no preâmbulo para que as minhas figuras estejam em português, e `\renewcommand{\refname}{Referências}` foi o que usei para renomear a mini bibliografia que inseri com `\begin{thebibliography}{99}... \end{thebibliography}`.

Ver <https://tex.stackexchange.com/a/82994/190644> para uma lista de comandos que interessam renomear.

Ignorar vários caracteres especiais com verbatim O pacote `verbatim` permite-nos ignorar uma série de caracteres especiais. Por exemplo, o comando `\verb|$_~|` produz `$_~`, e o comando `\verb+\verb|$_~|+` produz `\verb|$_~|`.

De modo semelhante, o ambiente `\begin{verbatim}` permite-nos fazer o mesmo mas ao longo de várias linhas. O conteúdo

```
Isto foi          escrito exa
tamente assim _{}|'\'^
```

foi escrito entre `\begin{verbatim}... \end{verbatim}`.

Inserir \ no texto Usa-se `\backslash`.

Lista de figuras e de tabelas com listofXXX `\listoffigures` produz a lista de figuras existentes:

List of Figures

1	Legenda da imagem	10
2	Legenda da imagem	10
3	Legenda principal	17
4	<code>\beginsubfigure[t]</code>	18
5	<code>\beginsubfigure[b]</code>	18

`\listoftables` funciona de modo semelhante.

Mudar a cor do texto com textcolor O comando `\textcolor{color}{text}` escreve o texto do segundo argumento com a cor indicada no primeiro argumento. A título de exemplo, **isto está a vermelho** foi escrito com `\textcolor{red}{isto está a vermelho}`.

As cores também podem ser especificadas **com o código RGB** — através do comando `\textcolor[RGB]{80,10,160}{com o código RGB}`.

Sublinhar texto com underline Texto sublinhado com `\underline{Texto sublinhado}`.

Destacar texto com o pacote soulutf8 Se incluirmos o pacote `soulutf8`, podemos usar o comando `\hl{}` para destacar texto com acentos. Por exemplo, **isto é importante!**.

2.3 Conteúdo avançado

Melhor coloração da sintaxe de código com o pacote minted **COMO FAZER ISTO?** ver https://en.wikibooks.org/wiki/LaTeX/Source_Code_Listings#The_minted_package e <https://github.com/gpoore/minted>

Separar o ficheiro em várias partes com subfiles

Gerir uma bibliografia com bibtex

Glossário, acrónimos e símbolos com glossaries

A Exemplo de secção no apêndice

As linhas de código imediatamente por cima desta linha de texto, no ficheiro `.tex`, são

```
\appendix
\section{Exemplo de secção no apêndice}
\label{sec:first-sec-in-appendix}
```