

# LaTeX

André Patacas, Afonso Boto



# LaTeX

André Patacas, Afonso Boto  
(93357) andrepatacas@ua.pt, (89285) afonso.boto@ua.pt

8 de Novembro de 2018 (Início)

## Resumo

Este relatório destina-se a explorar as funcionalidades da linguagem  $\text{\LaTeX}$ . Desde indexação de texto, inclusão de referências num ficheiro externo, criação de tabelas, introdução de símbolos matemáticos e até mesmo criar gráficos com pontos vetores e arrays destes. É um documento que foca na utilização prática da tecnologia e que visa introduzir pessoas novas à mesma. Apesar de já existirem manuais de extrema qualidade como [1] ou [2], geralmente pecam por se tornarem listas quase exaustivas de comandos enquanto o objetivo deste documento é apresentar a linguagem e dar sugestões de utilização como na Seção 3.7 em que é explorada a potencialidade do comando "`\input`".

Este documento não se propõe de forma alguma a substituir o manual de  $\text{\LaTeX}$ [3], é apenas um guião simples para iniciantes na linguagem e na forma de escrever textos, uma vez que a grande maioria conhece e utiliza ferramentas como Microsoft Word que utiliza o paradigma **What You See Is What You Mean** (WYSIWYM).

O documento tem algum ênfase nas funções de matemática por se tratarem de funções que produzem texto mais elaborado e com mais variantes que podem confundir utilizadores que acabam de ingressar na linguagem. Para além desta a maioria das funções mais gerais de  $\text{\LaTeX}$  serão abordadas ainda que com menos detalhe.

---

<sup>1</sup>Este relatório serve também como exercício prático dos conhecimentos adquiridos para a sua realização.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	História do $\text{\LaTeX}$	1
1.2	Motivação e Descrição Capítulos	1
<b>2</b>	<b>Introdução à estrutura <math>\text{\LaTeX}</math></b>	<b>3</b>
2.1	Filosofia	3
2.1.1	Prós	3
2.1.2	Contras	4
2.1.3	WYSIWYM	4
2.1.4	WYSIWYG	5
2.2	Estrutura de Comandos	5
<b>3</b>	<b>Aprofundamento da linguagem</b>	<b>6</b>
3.1	Pacotes $\text{\LaTeX}$	6
3.2	Caracteres Especiais	6
3.3	Inclusão de Imagens	7
3.4	Listas e Tabelas	9
3.4.1	Listas	9
3.4.2	Tabelas	10
3.5	Modo Matemática	11
3.5.1	Como utilizar o Modo de Matemática	11
3.5.2	Comandos mais comuns em $\text{\LaTeX}$	12
3.5.3	Caracteres do Modo Matemática	12
3.5.4	Frações	13
3.5.5	SuperScripts e SubScripts	13
3.5.6	Radicais	14
3.5.7	Integrais, Somatórios, Produtórios e Limites	14
3.5.8	Indexação de Equações	14
3.5.9	Matrizes	15
3.5.10	Chavetas	15
3.6	Criação Gráficos	16
3.7	Projetos com $\text{\LaTeX}$ Comando input	20
3.8	Referências a Documentos Externos	22
3.9	Fontes e Estilos de Texto	23

# Capítulo 1

## Introdução

### 1.1 História do L<sup>A</sup>T<sub>E</sub>X

A linguagem de descrição de texto L<sup>A</sup>T<sub>E</sub>X começou a ser desenvolvida com o nome de T<sub>E</sub>X no ano de 1977 e publicada em 1982 por Donald E. Knuth. L<sup>A</sup>T<sub>E</sub>X é uma linguagem que utiliza o código presente na linguagem T<sub>E</sub>X criando macros que facilitam a escrita de documentos. T<sub>E</sub>X foi criada com o intuito de tornar artigos de jornais e revistas mais atraentes ao público e eventualmente, com o surgimento do L<sup>A</sup>T<sub>E</sub>X tornou-se uma ferramenta utilizada mundialmente para realização de textos acadêmicos devido aos seus inúmeros recursos como indexação automática de texto, figuras, criação automática de índices, entre outros. Ainda a ressaltar duas curiosidades sobre a linguagem T<sub>E</sub>X, uma é a forma como é descrita a versão da mesma, a versão tende para  $\pi$  e está atualmente na versão 3.14159265. A outra é a forma como se pronuncia a palavra, a letra X é lida com som de "c" já que a letra grega chi " $\chi$ " tem esta pronúncia.

### 1.2 Motivação e Descrição Capítulos

O trabalho foi realizado com objetivo de aprofundar o conhecimento da linguagem L<sup>A</sup>T<sub>E</sub>X criando também um documento que facilite outros a conhecer melhor a linguagem.

Na Seção 2.1 está descrita a filosofia por detrás da linguagem como o facto de ser uma linguagem open source e uma markup language que implementa o conceito WYSIWYM.

Na Seção 2.2 é apresentada a estrutura fundamental dos comandos L<sup>A</sup>T<sub>E</sub>X que são fundamentais para a compreensão do funcionamento da linguagem.

Na Seção 3.1 está descrita a forma como é possível reaproveitar o código de terceiros para introduzir imagens, fórmulas matemáticas mais complexas, criar gráficos, entre outros.

Na Seção 3.2 será mostrado como introduzir caracteres especiais desde acentos (pouco comuns no inglês) até texto <sup>*superscript*</sup> recorrendo ao modo de mate-

mática.

Na Seção 3.3 é feita uma breve abordagem à inclusão de imagens externas no documento.

Na Seção 3.4 são descritas as tabelas em detalhe desde posicionamento do texto na tabela até à forma como estão dispostas as divisórias.

Na Seção 3.5 é apresentado em algum detalhe o modo de matemática que permite escrever fórmulas e símbolos matemáticos dos mais variados tipos.

Na Seção 3.6 há uma breve abordagem à criação de gráficos em  $\text{\LaTeX}$  mostrando o conceito e alguns comandos simples.

Na Seção 3.7 é apresentada uma forma de estruturar documentos  $\text{\LaTeX}$  em vários módulos mais pequenos.

Na Seção 3.8 é descrita a forma como podem ser apresentadas bibliografias e referências a outros documentos.

Na Seção 3.9 é apresentada a forma como se podem utilizar estilos e tamanhos de texto.

## Capítulo 2

# Introdução à estrutura L<sup>A</sup>T<sub>E</sub>X

### 2.1 Filosofia

Como qualquer editor de texto L<sup>A</sup>T<sub>E</sub>X tem as suas vantagens e desvantagens em relação a outros editores, portanto neste subcapítulo vamos enumerar algumas das suas vantagens e desvantagens e vamos comparar um pouco com o Word.

#### 2.1.1 Prós

**Velocidade e automatização:** Um arquivo tex é rápido, já que trata de texto apenas. O output, em pdf, é obviamente mais rápido do que Word. Links entre seções (também possíveis em Word) são facilmente implementados. O mesmo pode ser dito sobre figuras e gráficos, por exemplo.

**Layout e ferramentas científicas:** L<sup>A</sup>T<sub>E</sub>X usa um layout mais profissional. A interação entre diferentes tipos de objeto e o texto flui de forma excelente, algo raro em Word.

**Compatibilidade:** Diferentes versões de Word podem causar mudanças bruscas de formatação em doc. TeX, por outro lado, não sofre com esse tipo de problema. OK, pacotes são atualizados, e alguns detalhes podem mudar, mas o nível de compatibilidade é suficientemente baixo para que se possa simplesmente presumir que seja zero, principalmente quando comparamos com docs. Um tex pode ser aberto em virtualmente qualquer lugar. Pode-se editar mesmo sem ter LaTeX instalado, porque não é preciso compilar um tex para o editar: o resultado final (pdf) é um arquivo independente do arquivo com o conteúdo (tex).

**Pacotes:** LaTeX é gratuito (open source). Há milhares de pacotes disponíveis para as mais variadas tarefas. Diferentes desenvolvedores podem adicionar funcionalidades a partir de pacotes, essa é a grande vantagem de sistemas open source (há outras, é claro). O número de desenvolvedores/programadores no trabalho no Word é limitado, o que significa que bugs demoram mais a serem corrigidos. Em sistemas abertos, como a Wikipedia, há um número absurdo de pessoas a trabalhar constantemente para que a coisa toda funcione.

**Aprendizagem:** Usar LaTeX é aprender, constantemente, coisas diferentes. Se o utilizador nunca programou, utilizar LaTeX será uma introdução básica: a partir daí possivelmente poderá partir para outras linguagens depois. Isso porque o feedback é bastante instantâneo: aprende-se algo, compila-se, e percebe-se que conseguiu criar uma estrutura bastante complexa. Isso é estimulante, como qualquer atividade em que se aprende constantemente. Além disso, para pessoas que não são da área computação/programação, usar LaTeX é uma ótima oportunidade de "pensar" em códigos. Passa-se a entender uma nova sintaxe, e aplicar seus conhecimentos intuitivamente para criar aquilo que o Word não consegue. Isso é excelente não apenas para artigos/teses/dissertações, mas para handouts e apresentações de slides: muito do que usamos em Linguística exige uma certa complexidade gráfica, e transmitir isso nem sempre é intuitivo, principalmente em editores de texto como o Word.

### 2.1.2 Contras

Basicamente, a desvantagem de LaTeX é a interface. Se o utilizador tem pouco conhecimento na área e sabe apenas o básico, dificilmente irá gostar de usar LaTeX. É preciso paciência e dedicação no início, coisas que antes eram simples são, de repente, complicadas. Isso é comum a qualquer linguagem de programação.

Por outro lado, se o utilizador gosta de desenvolver diferentes habilidades e se for avançado na área, LaTeX é uma ótima opção. Este tipo de recurso está normalmente associado a estudantes que pretendam fazer progressão de estudos nomeadamente Mestrados e Doutoramentos. Nesse caso, certamente têm a capacidade para aprender uma linguagem como LaTeX. Ou seja, as pessoas que normalmente procuram LaTeX têm o nível de instrução necessário para a aprender, pois, as pessoas que mal sabem usar um computador não devem estar a pensar em publicar artigos científicos.

Se os utilizadores lidarem com pessoas que não utilizam LaTeX, nomeadamente em trabalhos de grupo, isso pode ser um problema. A solução é utilizar pdfs para comentários... o que talvez não seja ideal, dependendo do tipo de elementos que certo grupo tenha.

### 2.1.3 WYSIWYM

Esta expressão WYSIWYM em inglês significa, basicamente, que a exibição na tela deve primar pelas informações, e não pela formatação, que é um trabalho que deve ser deixado para o computador.

O que quer dizer que o utilizador escreve o conteúdo numa forma estruturada, marcando-o de acordo com o seu significado e a sua importância no documento, deixando a sua aparência final desde um ou mais estilos de folhas em separado.

A maior vantagem deste sistema é a separação total entre apresentação e conteúdo: os utilizadores podem estruturar e escrever o documento uma vez, em vez de constantemente alterá-lo cada modo de apresentação.



### 2.1.4 WYSIWYG

Num documento **What You See Is What You Get** (WYSIWYG) implica que uma interface do utilizador permite que o utilizador veja o produto final muito semelhante enquanto o documento está a ser criado. No geral, isto implica a habilidade direta de manipular um layout de um documento sem ter de escrever ou lembrar de nomes de comandos de layout. O significado real depende da perspectiva do utilizador.

## 2.2 Estrutura de Comandos

Um arquivo fonte do  $\text{\LaTeX}$  contém, além do texto a ser processado, comandos que indicam como o texto deve ser processado. Palavras são separadas por um ou mais espaços. Parágrafos separados por uma ou mais linhas em branco. A saída não é afetada por espaços extra ou por linhas em branco extras. A maioria dos comandos do  $\text{\LaTeX}$  são iniciadas com o carácter " $\backslash$ ". **Reginaldo** Uma " $\backslash$ " sozinha produz um espaço. Um ambiente é uma região do texto que tem um tratamento especial. Um ambiente é iniciado com

$\backslash\text{begin}\{\text{nome do ambiente}\}$  e terminado por  $\backslash\text{end}\{\text{nome do ambiente}\}$ .

O normal padrão de comandos são do tipo  $\backslash\text{comando}\{\text{argumento}\}[\text{Modifier}]$  ou  $\backslash\text{comando}\{\text{argumento}\}$  ou até apenas  $\backslash\text{comando}$ .

Exemplos:

$$\backslash\text{sqrt}[3]\{x\} \rightarrow \sqrt[3]{x}$$

$$\backslash\text{sqrt}\{x\} \rightarrow \sqrt{x}$$

```
$
f(x) = \begin{cases}
1 & \text{\& \text{ } \text{se } x < 2} \\
b & \text{\& \text{ } \text{se } x \leq 2}
\end{cases}
$
```

Gera:

$$f(x) = \begin{cases} 1 & , \text{ se } x < 2 \\ b & , \text{ se } x \leq 2 \end{cases}$$

## Capítulo 3

# Aprofundamento da linguagem

### 3.1 Pacotes $\text{\LaTeX}$

O  $\text{\LaTeX}$  define um conjunto básico de macros para edição de textos. Caso o utilizador quera usar alguma função mais complexa, o  $\text{\LaTeX}$  permite que ele inclua arquivos com novos macros. Esses arquivos são chamados de pacotes. Existem pacotes para escrever a cor , para incluir figuras , etc ... O utilizador pode ainda até criar o seu próprio pacote.

Para incluir um pacote basta usar :

```
\usepackage[opção]{nome do pacote}
```

Alguns pacotes podem ser incluídos usando opções diferentes . A opção deve ser inserida entre parêntesis retos antes do nome do pacote. Neste exemplo :

```
\usepackage[portuguese]{babel}
```

o pacote babel define macros para edição de textos em diversas línguas . Como queríamos escrever em Português pusemos a opção portuguese. As distribuições do  $\text{\LaTeX}$  costumam vir com um conjunto amplo de pacotes. Outros pacotes podem ser instalados da mesma forma.

### 3.2 Caracteres Especiais

Em  $\text{\LaTeX}$  alguns caracteres possuem funções especiais, que normalmente associamos a comandos. A estes caracteres, que normalmente não fazem parte do conteúdo do documento, chamamos de caracteres especiais pois assumem um papel essencial no código fonte de um documento  $\text{\LaTeX}$ , estes são:

& \$ # % \_ { } ^ ~ e \

Eles são impressos com os comandos:

`\& \$ \% \_ \{ \} \^{} \~{} e \textbackslash`

**character "\$" :**

Este caracter é usado para inicializar ou terminar o modo matemática .

**character "\"**

este caracter é o caracter base de qualquer comando, está no inicio de qualquer:

`\'{a}` → á

`\LaTeX` → L<sup>A</sup>T<sub>E</sub>X

**Character "%":**

Este caracter é usado para fazer comentários , ou seja, vai ser ignorado pelo compilador de L<sup>A</sup>T<sub>E</sub>X.

**Character "~":**

Este caracter serve para dizer ao compilador que 2 palavras nunca devem ficar separadas em linhas consecutivas mas sim na mesma linha.

### 3.3 Inclusão de Imagens

Imagens são elementos essenciais em quase todos os documentos científicos. O L<sup>A</sup>T<sub>E</sub>X providencia várias opções para manipular imagens e fazê-las parecer exatamente como o utilizador precisa, no entanto, o L<sup>A</sup>T<sub>E</sub>X não possui nenhum mecanismo nativo de inclusão de imagens, temos que recorrer ao uso de packages. **labi** Neste subcapítulo vamos explicar como incluir imagens nos formatos mais comuns , como encolhe-las , como as aumentar , como as rodar, e ainda como as referenciar nos documentos.

Vamos usar como exemplo o uso do logotipo da UA.

Aqui - `\includegraphics[scale=0.5]{ua.pdf}`  
- o logotipo da UA.

que origina o resultado :



Aqui - o logotipo da UA.

Também é possível utilizar esta imagem dentro de um objeto flutuante:

```
\begin{figure}[h]
\center %imagem centrada
\includegraphics[scale=0.5]{ua.pdf}
\caption{Logotipo da UA} %legenda
\label{fig:ualogo.1}
\end{figure}
```



Figura 3.1: Logotipo da UA

Na inclusão de um ficheiro é usual indicar o seu nome sem extensão (nos casos acima o nome completo do ficheiro é ua.pdf e está na pasta do próprio documento). Com efeito, podem existir vários ficheiros para a mesma imagem, cada um com o seu formato, e deste modo facilita-se a compilação do documento para formatos de saída diferentes. O comando de inclusão de imagens possui várias opções, entre as quais as que permitem redimensionar ou de outra forma ajustar a imagem a incluir:

**height** - altura da imagem.

**width** - largura da imagem.

**scale** - fator de escala.

**angle** - ângulo de rotação.

Mais a baixo, na figura 3.2 vamos mostrar vários exemplos uteis de como a inclusão de imagem funcionam:

```
\begin{figure}[h]
\center % Centra as imagens
a) \includegraphics{ua.pdf}
b) \includegraphics[height=2cm]{ua.pdf}
c) \includegraphics[width=10mm]{ua.pdf}
d) \includegraphics[scale=.5,angle=90]{ua.pdf}
e) \includegraphics[height=5mm,width=2cm]{ua.pdf}
\caption{Logotipo da Universidade de Aveiro: a) na dimensão real,
b) com 2cm de altura, c) com 10mm de largura, d) com altura e largura
reduzidas a  $1/2$  e simultaneamente rodado  $90^\circ$  e e) com uma modificação
anamórfica da altura e da largura.}
\label{fig:ualogo.2}
\end{figure}
```

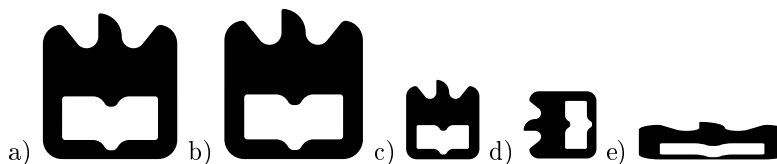


Figura 3.2: Logotipo da Universidade de Aveiro: a) na dimensão real, b) com 2cm de altura, c) com 10mm de largura, d) com altura e largura reduzidas a  $1/2$  e simultaneamente rodado  $90^\circ$  e e) com uma modificação anamórfica da altura e da largura.

## 3.4 Listas e Tabelas

O  $\text{\LaTeX}$  oferece a possibilidade de criar listas e tabelas para facilitar a leitura e compreensão do texto.

### 3.4.1 Listas

Existe dois tipos de listas em  $\text{\LaTeX}$ , listas ordenadas e não ordenadas. Para criação de listas não ordenadas utiliza-se o comando "`\begin{itemize} ... \end{itemize}`" juntamente com o comando "`\item`":

- item0
- item1
- item2

Gerado com o código:

```
\begin{itemize}
\item item0
\item item1
\item item2
\end{itemize}
```

Para criação de listas ordenadas utiliza-se o comando "`\begin{enumerate} ... \end{enumerate}`" juntamente com o comando "`\item`":

1. item0
2. item1
3. item2

Gerado com o código:

```
\begin{enumerate}
\item item0
\item item1
\item item2
\end{enumerate}
```

Podem criar-se tabelas aninhadas que, caso numeradas, criam numeração automática das tabelas internas com um símbolo diferente:

1. item0
  - (a) item00
  - (b) item01
- i. item010

- ii. item111
  - iii. item212
- (c) item02
  - item020
  - item121
  - item222
- 2. item2

Gerado com o código:

```
\begin{enumerate}
\item item0
\begin{enumerate}
\item item00
\item item01
\begin{enumerate}
\item item010
\item item111
\item item212
\end{enumerate}
\item item02
\begin{itemize}
\item item020
\item item121
\item item222
\end{itemize}
\end{enumerate}
\item item2
\end{enumerate}
```

### 3.4.2 Tabelas

As tabelas em  $\text{\LaTeX}$  são um comando ligeiramente mais complexo que as listas. Uma tabela simples começa com o comando "`\begin{table}[mod] ... \end{table}`". Este modificador geralmente é "h" para "here", indicando que a tabela vai ser posta, se possível, no lugar relativo ao código que a descreve. No interior deste comando é inserido outro "`\begin{tabular}{tipo}\end{tabular}`", seja o tipo a forma como os itens estão dispostos horizontalmente, com ou sem linha vertical a separá-los, alinhados ao centro (c), à esquerda (l) ou à direita (r). Para além destes é ainda possível decidir quando existe uma quebra de linhas com o comando "`\\`" e se existirá uma linha horizontal entre as duas linhas com o comando "`\hline`", itens são separados por um e comercial "& ":

Gerado com o código:

	0	1
0	00	10
1	01	11

```

\begin{table}[h]
\center
\begin{tabular}{|c|l|r}
\hline
& 0 & 1 \\
\hline
0 & 00 & 10 \\
\hline
1 & 01 & 11 \\
\hline
\end{tabular}
\end{table}

```

## 3.5 Modo Matemática

O modo de matemática permite escrever símbolos e fórmulas de matemática de forma explícita e não ambígua que após compilados se assemelham extremamente a forma escrita dos mesmos

### 3.5.1 Como utilizar o Modo de Matemática

Existem dois modos de escrever matemática em  $\text{\LaTeX}$ , um é o Inline Math Mode (ILMM) que é iniciado terminado por um único cifrão, "\$", e que vai escrever a fórmula na linha onde é escrita reduzindo a formatação de texto para que encaixe na linha, o outro é o Newline Math Mode (NLMM) que inicia e termina com um cifrão duplo "\$\$" e vai criar a fórmula centrada numa nova linha com toda a formatação possível. Pode ainda usar-se o comando "\begin{equation} ... \end{equation}", este comando será discutido na Subseção 3.5.8.

Por exemplo no ILMM " $\int\limits_a^b f(x)$ " apareceria da seguinte forma: " $\int_a^b f(x)$ "

enquanto que no NLMM seria:

$$\int_a^b f(x)$$

Existe ainda uma forma de utilizar o ILMM com a formatação completa, o comando " $\displaystyle\int_a^b f(x)$ ", o que por vezes pode criar grandes espaçamentos entre linhas. Por exemplo o comando " $\displaystyle\int_a^b f(x)$ "

tornar-se-ia " $\int_a^b f(x)$ "

### 3.5.2 Comandos mais comuns em L<sup>A</sup>T<sub>E</sub>X

Alguns dos comandos mais comuns do modo de matemática do L<sup>A</sup>T<sub>E</sub>X são

Código	ILMM	displaystyle ILMM
<code>\sqrt{x}</code>	$\sqrt{x}$	$\sqrt{x}$
<code>x^p</code>	$x^p$	$x^p$
<code>x_b</code>	$x_b$	$x_b$
<code>x^p_b</code>	$x_b^p$	$x_b^p$
<code>\pi</code>	$\pi$	$\pi$
<code>\otimes</code>	$\otimes$	$\otimes$
<code>\cup</code>	$\cap$	$\cap$
<code>\subset</code>	$\subset$	$\subset$
<code>\sum</code>	$\Sigma$	$\Sigma$

### 3.5.3 Caracteres do Modo Matemática

Em matemática regularmente é necessária a introdução de alguns caracteres diferentes dos American Standard Code for Information Interchange (ASCII) que temos no teclado. Geralmente estes são letras gregas que podem ser obtidas pelo seu nome em inglês, no geral, pelo comando "`\nome`" para letras minúsculas e pelo comando "`\Nome`" para letras maiúsculas.

Exemplificando:

Código	Output
<code>\phi</code>	$\phi$
<code>\Phi</code>	$\Phi$
<code>\delta</code>	$\delta$
<code>\Delta</code>	$\Delta$

Podem também introduzir-se caracteres como infinito e "tender para" com os comandos "`\infty`" ( $\infty$ ) e "`\to`" ( $\rightarrow$ ).

Em matemática uma forma comum de representar somatórios é com uma soma incluindo 3 pontinhos no meio da fórmula:

$$\sum_{i=0}^n i = 1 + 2 + 3 + \cdots + n$$

Estes 3 pontinhos podem ser obtidos com o comando "`\dots`". Variantes deste símbolo com pontos na vertical e diagonal para matrizes por exemplo conseguem-se com as variações "`\vdots`"  $\vdots$ , "`\ddots`"  $\ddots$  e "`\reflectbox{$\ddots$}`"  $\ddots$ .

Podem representar-se desigualdades como o comando "`\tipo`" seja o tipo as iniciais da desigualdade em inglês:





### 3.5.6 Radicais

Em  $\text{\LaTeX}$  radicais são criados com a função " $\sqrt[n]{x}$ ". Por exemplo:

$$\sqrt[3i]{x^2+1}$$

Gerado com o código:

```
$$ \sqrt[3i]{x^2+1} $$
```

### 3.5.7 Integrais, Somatórios, Produtórios e Limites

O uso de integrais e somatórios em  $\text{\LaTeX}$  tem as suas particularidade e pode variar com de acordo com gosto pessoal. A forma mais simples de representar um integral, um somatório, um produtório e um limite é com os comandos " $\int$ ", " $\sum$ ", " $\prod$ " e " $\lim$ ".

Para além destas existem também as formas compostas com limites superiores e inferiores que podem ser representadas de duas formas, utilizando os já vistos superscript e subscript ou um novo comando chamado " $\lims^{\_}$ ". O comando " $\displaystyle$ " também afeta a forma como é apresentada a fórmula. A única diferença entre os dois comandos é a forma de representação final.

Ao utilizar os comandos de superscript e subscript temos:

$$\int_a^b f(x)dx = \lim_{||\Delta x|| \rightarrow 0} \sum_{i=1}^n f(x_i^*) \Delta x_i$$

Gerado com o código:

```
$$\int_a^b f(x)dx=\lim_{||\Delta x||\to 0}\sum_{i=1}^nf(x_i^*)\Delta x_i$$$
```

Com os comandos de limite temos:

$$\int_a^b f(x)dx = \lim_{||\Delta x|| \rightarrow 0} \sum_{i=1}^n f(x_i^*) \Delta x_i$$

Gerado com o código:

```
$$\int \lims^b_a f(x)dx=\lim \lims_{||\Delta x||\to 0}\sum \lims_{i=1}^nf(x_i^*)\Delta x_i$$$
```

### 3.5.8 Indexação de Equações

O modo de matemática do  $\text{\LaTeX}$ , munido da biblioteca  $\mathcal{AMS}\text{-}\text{\LaTeX}$ , oferece também a possibilidade de referência a equações através do comando " $\eqref{ref}$ ". Para isso é necessário criar uma secção de texto com o comando " $\begin{equation} \dots \end{equation}$ ":

$$\sum_{i=1}^{\infty} \tag{3.1}$$

Gerado com o código:

```
\begin{equation}
\sum_{i=1}^{\infty} \label{somatório}
\end{equation}
```

Pode referenciar-se a equação pelo seu label com o comando "`\eqref{somatório}`", aparecendo no documento final desta forma (3.1)

Para mudar a forma como a equação é numerada é possível utilizar o comando "`\tagtag`" que substitui a numeração pelo conteúdo da tag como no exemplo:

$$\sum_{i=1}^{\infty} \quad (\text{somatorio})$$

Gerado com o código:

```
\begin{equation}
\sum_{i=1}^{\infty} \label{somatorio2} \tag{somatório}
\end{equation}
```

Será referenciado na mesma pelo comando "`\eqref{somatorio2}`" desta forma (somatorio). É necessário ainda o cuidado com a ordem do label e da tag que têm de ser explicitamente label  $\rightarrow$  tag.

### 3.5.9 Matrizes

Para representar matrizes em  $\text{\LaTeX}$  utiliza-se o comando "`\begin{bmatrix}`" ... "`\end{bmatrix}`" e é uma estrutura bastante semelhante às listas:

$$\begin{bmatrix} a & b & c \\ d & \ddots & \vdots \\ e & \dots & f \end{bmatrix}$$

Gerado com o código:

```
\[
\begin{bmatrix}
a & b & c \\
d & \ddots & \vdots \\
e & \dots & f
\end{bmatrix}
\]
```

### 3.5.10 Chavetas

Em  $\text{\LaTeX}$  existem vários tipos de chavetas, a mais simples é a chaveta utilizada para representar sistemas de equações com o comando "`\begin{cases}`" ... "`\end{cases}`".

$$f(x) = \begin{cases} 1 & , \text{ se } x < 2 \\ b & , \text{ se } x \leq 2 \end{cases}$$

Gerado com o código:

```
$
f(x) = \begin{cases}
1 & \& \text{ , se } x < 2 \\
b & \& \text{ , se } x \leq 2 \\
\end{cases}
$
```

Podem ainda usar-se chavetas em cima e em baixo de equações com os comandos "`\underbrace`" e "`\overbrace`":

$$\overbrace{a+a}^{2a} + \underbrace{b+b}_{2b} = 2a + 2b$$

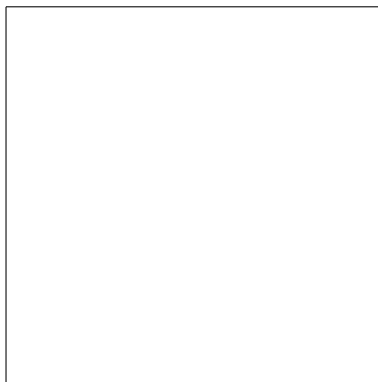
Gerado com o código:

```
$$ \overbrace{a+a}^{2a} + \underbrace{b+b}_{2b} = 2a + 2b $$
```

## 3.6 Criação Gráficos

O ambiente  $\text{\LaTeX}$  oferece a possibilidade de criar gráficos na própria linguagem sem necessidade de incluir imagens externas de outros sites por exemplo Geogebra ou Desmos.

Para iniciar um ambiente de gráfico ortonormado de coordenadas entre x e y insere-se o comando "`\begin{picture}(x,y) ... \end{picture}`". Para que o gráfico seja visível em unidades predefinidas pode utilizar-se o comando "`\setlength\unitlength{tamanho}`" com por exemplo 5cm. Por fim o comando para desenhar um linha é definido pelo vetor origem, o vetor direção e o tamanho da linha. O comando para instanciar a linha é "`\put(VetorOrigem){\line (VetorDireção){tamanhoDaLinha}}`". Exemplificando a composição dos comandos descritos:



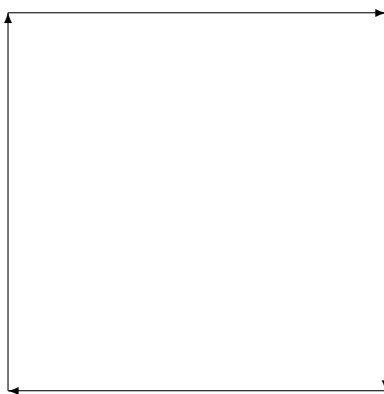
Gerado com o código:

```
\begin{center}

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
\put(0,0){\line(0,1){1}}
\put(1,0){\line(-1,0){1}}
\put(0,1){\line(1,0){1}}
\put(1,1){\line(0,-1){1}}
\end{picture}

\end{center}
```

Esta é a forma mais simples de criação de gráficos em  $\text{\LaTeX}$ , e tem as suas variações por exemplo com o comando "`\vector`" em vez do "`\line`", que vai criar uma variação com setas em vez de linhas:



Gerado com o código:

```
\begin{center}
```

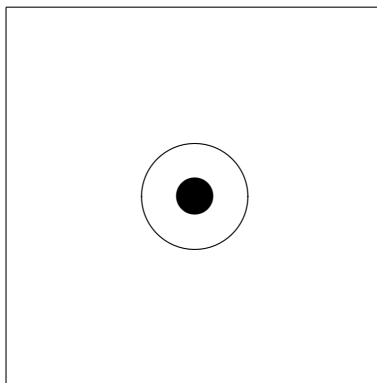
```

\setlength{\unitlength}{5cm}
\begin{picture}(1,1)
\put(0,0){\vector(0,1){1}}
\put(1,0){\vector(-1,0){1}}
\put(0,1){\vector(1,0){1}}
\put(1,1){\vector(0,-1){1}}
\end{picture}

\end{center}

```

Podem criar-se circunferências com o comando "`\circle{raio}`" e círculos com o comando "`\circle*{raio}`":



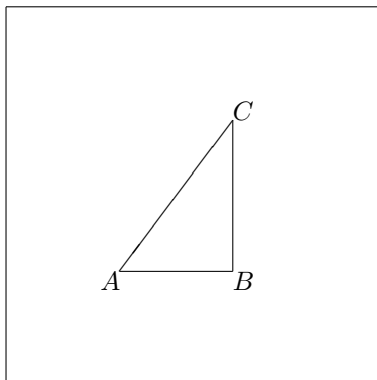
Gerado com o código:

```

\begin{center}
\setlength{\unitlength}{5mm}
\begin{picture}(10,10)
%desenhar quadrado
\put(0,0){\line(0,1){10}}
\put(10,0){\line(-10,0){10}}
\put(0,10){\line(10,0){10}}
\put(10,10){\line(0,-10){10}}
%desenhar Círculos
\put(5,5){\circle{10}}
\put(5,5){\circle*{1}}
\end{picture}
\end{center}

```

É possível também escrever textos nos gráficos gerados, assim, por exemplo um triângulo pode ter vértices nomeados. O comando para tal efeito é "`\put(Origem){$texto$}`". Exemplificando:

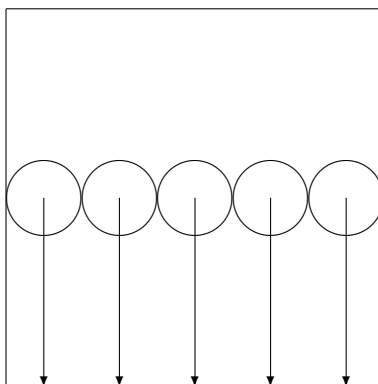


Gerado com o código:

```
\begin{center}
\setlength{\unitlength}{5mm}
\begin{picture}(10,10)
%desenhar quadrado
\put(0,0){\line(0,1){10}}
\put(10,0){\line(-10,0){10}}
\put(0,10){\line(10,0){10}}
\put(10,10){\line(0,-10){10}}
%desenhar triângulo
\put(3,3){\line(1,0){3}}
\put(6,3){\line(0,1){4}}
\put(6,7){\line(-3,-4){3}}
%Letras Vértices
\put(2.5,2.5){$A$}
\put(6,2.5){$B$}
\put(6,7){$C$}
\end{picture}

\end{center}
```

Para além destas existe também a funcionalidade de escrever arrays de objetos como linhas ou círculos com o comando "`\multiput(x,y)( $\Delta x$ , $\Delta y$ )n\{objeto\}`". Por exemplo:



Gerado com o código:

```
\begin{center}

\setlength{\unitlength}{5mm}
\begin{picture}(10,10)

%desenhar quadrado
\put(0,0){\line(0,10){10}}
\put(10,0){\line(-10,0){10}}
\put(0,10){\line(10,0){10}}
\put(10,10){\line(0,-10){10}}
%Desenhar Array de Vetores
\multiput(1,5)(2,0){5}{\vector(0,-1){5}}
%Desenhar Array de Círculos
\multiput(1,5)(2,0){5}{\circle{2}}
\end{picture}
\end{center}
```

### 3.7 Projetos com $\text{\LaTeX}$ Comando input

Na realização de projetos com  $\text{\LaTeX}$  é comum a situação de vários integrantes do projeto quererem editar o documento e interagirem com uma plataforma como o git. Neste caso, não podem, simultaneamente, editar o mesmo ficheiro ao mesmo tempo, o que cria uma necessidade de separar o documento em partes. Esta separação pode ser feita através do comando "`\input{filepath/filename}`". Este comando funciona de forma semelhante à instrução de compilador "`#include`" em C++ copiando o conteúdo do ficheiro passado como argumento no sítio onde é invocada a função input. Um exemplo de uma separação seria criar um ficheiro para cada capítulo do livro/apresentação/tese e distribuir tarefas de maneira que os integrantes do projeto editassem apenas um documento de cada vez.



Exemplificando com este documento, no documento principal é utilizado o comando "\input{" para importar outro ficheiro que contém a informação por capítulos:

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%documento principal
\chapter{Introdução: História do \LaTeX}
\label{chap.introducao}
\input{textos/introducao.tex}

\chapter{Introdução à estrutura \LaTeX}
\input{textGenScripts/chapter.tex}

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%documento de subcapítulos
"textGenScripts/chapter.tex"
...
\section{Filosofia}
\label{chap.filosofia}
\input{textos/filosofia.tex}

\section{Estrutura de Comandos}
\label{chap.estruturadecomandos}
\input{textos/estruturadecomandos.tex}

\section{Projetos com \LaTeX Comando input}
\label{chap.projetoscomlatexcomandoinput}
\input{textos/projetoscomlatexcomandoinput.tex}
...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%exemplo documento do subcapítulo em si
"textos/projetoscomlatexcomandoinput.tex"
Na realização de projetos com \LaTeX é comum a situação de
vários integrantes do projeto quererem editar o documento e
interagirem com uma plataforma como o git. Neste caso, não
podem, simultaneamente, editar o mesmo ficheiro ao mesmo tempo,
o que cria uma
...
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Uma ideia de utilização deste comando é a criação de um pequeno script para gerar o código repetitivo automaticamente dados os nomes dos capítulos. Na realização deste relatório foi utilizado python3 com esse objetivo:

```
file1 = open('chapter.tex','w')
```

```

for i in ['Estrutura de Comandos','Criação Macros','Pacotes
\LaTeX','Caracteres Especiais', 'Listas e Tabelas','Disposição
e Organização de Texto','Modo Matemática','Criação Gráficos',
'Projetos com \LaTeX: Comando input','Referências a Documentos
Externos', 'Inclusão de Imagens','Fontes e Linguagens']:
    a = i.replace(' ','').lower().replace('ç','c').replace('ã',
    'a').replace('á','a').replace(':',',').replace('\\','').
    replace('ê','e')
    b = ""
    \section{""+i+""}
    \label{chap.""+a+""}
    \input{textos/""+a+"".tex}\n""
    file1.write(b)
    file2 = open('../textos/' + a + '.tex','w')
    file2.write("Escrever aqui Capitulo " + i)
    file2.close()
file1.close()

```

### 3.8 Referências a Documentos Externos

Devido ao seu cariz principalmente académico,  $\text{\LaTeX}$  suporta a inclusão de referências a bibliografia, utilizando o comando de referência "`\cite{nome}`" em conjunto com o comando de definição "`misc{nome, atributos}`", como por exemplo:

```

%comando de definição
@misc{tnssl,
    author={{Tobias Oetiker}},
    title={{The Not So Short Introduction to \LaTeX}},
    month={Março},
    year={2018},
    note = "Online; acedido em Novembro de 2018]"
}

%comando de citação
\cite{tnssl}

```

O resultado obtido com a citação é semelhante a este [1]

Podemos incluir as definições num documento externo através do comando "`\input{}`" (aprofundado na Seção 3.7) mas o mais comum é criar um ficheiro com a extensão ".bib" e utilizar o comando "`\bibliography{filename}`" para referenciar que é um ficheiro de bibliografia.

Utiliza-se o comando "`\printbibliography`" para indicar ao compilador onde deve ser escrita a bibliográfica caso referenciada.

### 3.9 Fontes e Estilos de Texto

L<sup>A</sup>T<sub>E</sub>X oferece a possibilidade de alterar o tamanho, o estilo. Infelizmente não é possível utilizar fontes externas nativamente no L<sup>A</sup>T<sub>E</sub>X para tal efeito recomenda-se a utilização de XeL<sup>A</sup>T<sub>E</sub>X.

A mudança de estilo e tamanho consegue-se através dos comandos listados de seguida:

Código	Resultado
<code>\textrm{textrm}</code>	textrm
<code>\textsf{textsf}</code>	<b>textsf</b>
<code>\texttt{texttt}</code>	<b>texttt</b>
<code>\textmd{textmd}</code>	textmd
<code>\textbf{textbf}</code>	<b>textbf</b>
<code>\textup{textup}</code>	textup
<code>\textit{textit}</code>	<i>textit</i>
<code>\textsl{textsl}</code>	<i>textsl</i>
<code>\textsc{textsc}</code>	TEXTSC
<code>\emph{emph}</code>	<i>emph</i>
<code>\textnormal{textnormal}</code>	textnormal

Código	Resultado
<code>\tiny{tiny}</code>	<small>tiny</small>
<code>\scriptsize{scriptsize}</code>	<small>scriptsize</small>
<code>\footnotesize{footnotesize}</code>	<small>footnotesize</small>
<code>\small{small}</code>	<small>small</small>
<code>\normalsize{normalsize}</code>	<small>normalsize</small>
<code>\large{large}</code>	<small>large</small>
<code>\Large{Large}</code>	<small>Large</small>
<code>\LARGE{LARGE}</code>	<small>LARGE</small>
<code>\huge{huge}</code>	<small>huge</small>
<code>\Huge{Huge}</code>	<small>Huge</small>

# Contribuições dos autores

O documento tem 2 autores e como tal, as contribuições de cada um. O autor André Patacas (AP) foi responsável pela estruturação do documento e toda a pesquisa e escrita relativa ao resumo e aos capítulos 1.1, 1.2 e 3.4 até 3.9 incluindo o capítulo Seção 3.5. O coautor do documento Afonso Boto (AB) esteve a cargo dos restantes capítulos, 2.1 a 3.4.

# Acrónimos

**ILMM** Inline Math Mode

**NLMM** Newline Math Mode

**ASCII** American Standard Code for Information Interchange

**WYSIWYG** What **Y**ou **S**ee **I**s What **Y**ou **G**et

**WYSIWYM** What **Y**ou **S**ee **I**s What **Y**ou **M**ean

**AP** André Patacas

**AB** Afonso Boto

# Bibliografia

- [1] Tobias Oetiker, *The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X*, [Online; acedido em Novembro de 2018], mar. de 2018.
- [2] (Desconhecido), Universidade de Aveiro, *Produção de documentos com L<sup>A</sup>T<sub>E</sub>X*, [Online; acedido em Novembro de 2018], 2018-2019.
- [3] L. Lamport, *LaTeX: A Document Preparation System*, 2nd. Addison-Wesley Professional, 1994, p. 288.