

A minha experiência em \LaTeX

Antero Neves

Eletrônico anteroneves.reg@gmail.com

Liame <http://aprendolatex.wordpress.com/>

Resumo Este artigo tem dois objectivos claros. O primeiro é partilhar a minha experiência com o \LaTeX no que diz respeito a resultados e conhecimentos, e o segundo, consequência do primeiro, é facilitar o começo daqueles que agora se iniciam neste mundo.

Embora seja impossível descrever aqui todos as potencialidades presentes no \LaTeX apresentam-se aquelas que julgo serem as básicas para qualquer pessoa poder começar e procuro também responder às dúvidas, que em minha opinião são mais recorrentes nos principiantes.

Para uma melhor compreensão do texto, estão presentes alguns exemplos de aplicação específica assim como de uma estrutura completa de um documento \LaTeX .

Espero que com este artigo consiga dar o meu contributo para o crescimento de uma comunidade que sempre me apoiou, ajudando a eliminar o espectro da dificuldade que teima em reinar sobre o mundo do \LaTeX .

1 Introdução

A mudança sempre nos faz pensar e quando se faz para melhor, aí é que devemos pegar com as duas mãos! É isso que eu digo a todos que querem se aventurar neste mundo que é o \LaTeX . A verdade é que a dificuldade não está em saber \LaTeX , mas sim em ultrapassar a diferença que existe entre \LaTeX e outros.

Não podemos encarar a edição de um texto em \LaTeX da mesma maneira como encaramos no MS Word pura e simplesmente porque diferem na forma de como são feitas, pois têm abordagens diferentes, têm procedimentos diferentes e claro, têm resultados diferentes - ainda bem! :-)

Com este artigo, eu venho mostrar a todos que o \LaTeX é muito simples e é muito mais fácil do que aparenta à primeira vista. Primeiro, porque os procedimentos não são assim tantos como dizem as más línguas, nem tão difíceis

de entender como dizem os humanistas. Segundo porque, para além de um conjunto de livros que podemos adquirir ou encontrar em bibliotecas, existe um mar de documentação livre de custos na internet e também muitas pessoas que estão dispostas a nos ajudar. E em terceiro lugar, temos que contar também com a nossa capacidade natural de experimentar alterações de forma a corrigir erros que possam aparecer (adaptação).

Por outro lado, o uso desta tecnologia reflecte-se também na nossa conta bancária, uma vez que podemos adquirir o necessário para trabalhar a custo 0 e não precisamos mudar de computador, ou fazer upgrades de hardware dispendiosos para acompanhar a evolução do software.

2 Meu primeiro contacto com L^AT_EX

Comecei a escrever em L^AT_EX em 2005 e devo confessar que o fiz com um sentimento misto de obrigatoriedade e curiosidade. Obrigatoriedade, porque tinha que entregar a minha monografia de final de curso – Matemática – escrita em L^AT_EX. Curiosidade, porque embora nunca tivesse feito nenhum documento em L^AT_EX, já havia tido algumas conversas com colegas do meu curso e também de outros cursos ligados à informática. Todos me diziam maravilhas do “lah-teque-se” – era assim que lhe chamávamos na altura, só mais tarde descobri que se deveria dizer “Lay-Tech”.

Um outro factor que fazia com que o L^AT_EX estivesse nas minhas boas graças era o facto de eu conseguir tirar melhores notas em todas as disciplinas que estudava com o apoio de documentos feitos em L^AT_EX. Isto é um facto!

Como diz o ditado: *os olhos também comem!* Ninguém gosta de estudar por documentos sem apresentação e isto nota-se muito na área da matemática onde é preciso apresentar imensos símbolos que, em documentos escritos por outros métodos, ficam com uma aparência . . . horrível!

Mas este artigo destina-se a futuros escritores, e quando concebemos um documento, umas das questões que nos assalta é: será que os meus leitores vão achar isto agradável à vista? Julgo ser consensual que um documento produzido em L^AT_EX tenha uma aparência muito mais profissional, o que pode ajudar qualquer um a tirar uma nota superior àquela que tiraria se apresentasse o mesmo trabalho feito de outra forma, e disso sou testemunha!

Por isso pensem bem, pode ser um pouco difícil à partida, porque é algo novo

e diferente, mas poderão perceber rapidamente esse efeito de apreciação positiva em termos estéticos. Esta é a vantagem mais imediata que encontro no \LaTeX – a estética.

Depois de acabado o curso, ingressei num mestrado, também na área da Matemática, onde encontrei gente de todas as tribos, quase todos mais velhos que eu e a maior parte nunca tinham sequer visto um editor de \LaTeX à frente, pois continuavam a usar as ferramentas comuns para fazer os seus trabalhos e mantinham aqueles problemas típicos, tais como problemas de formatação de fontes, de margens, de tamanho do papel, colocação de imagens, tamanho enorme dos ficheiros, compatibilidade entre versões e por aí adiante. Sempre que mudavam de computador tinham que ter tudo isso em atenção e alterar o que fosse necessário – se possível.

É verdade que hoje existem impressoras virtuais que produzem ficheiros *.pdf*, mas é um facto que o tamanho não se torna assim tão reduzido e continuamos com um problema grave de possibilidade de edição. Por exemplo, quando queremos enviar um texto a alguém para ser corrigido, aumentado ou editado a qualquer nível. É realmente terrível!

E sim, é também verdade que o documento final do \LaTeX pode ser também um ficheiro *.pdf*, mas temos duas situações distintas: a primeira é o trabalho final, que podemos apresentar no referido *.pdf*, a segunda é que queremos que alguém edite o documento e para isso temos o ficheiro *.tex* gerado pelo editor de \LaTeX .

Por isso, agora que estou numa fase adiantada do mestrado, uma das minhas tarefas tem sido introduzir os meus colegas ao \LaTeX e também como eu, eles estão um pouco contrariados, mas é início e já serão rendidos às suas vantagens.

3 Por onde começar

3.1 Pré-requisitos

Como tudo na vida, o mais importante é compreender como as formas se encaixam e como as coisas funcionam.

Primeiro é preciso saber que no \LaTeX estão envolvidos dois processos, o da edição do texto e o da compilação, isso implica a existência de um editor e um compilador.

O que se faz no editor?

Um editor de texto – compreenda-se de \LaTeX – serve para colocar todo o texto que queremos que apareça no trabalho e, para além disso, colocar todas as informações de formatação necessárias. As formatações e outras especificações são fornecidas por comandos.

Hoje há vários editores de \LaTeX e muitos são freeware, ou seja, podemos usa-los livremente sem pagar, o que representa uma enorme vantagem!

Programa	Sistema Operativo	Tipo	Site na internet
TeXnicCenter	Windows	Freeware	http://www.toolscenter.org/
TeXmaker	Vários	Freeware	http://www.xmlmath.net/texmaker/
LEd	Windows	Freeware	http://www.latexeditor.org/
Kile	Linux	Freeware	http://kile.sourceforge.net/
Lyx	Linux	Freeware	http://www.lyx.org/
Vim	Vários	Freeware	http://www.vim.org/
WinEdt	Windows	Shareware	http://www.winedt.com/

Table 1: Alguns editores de \LaTeX

O que faz um compilador?

O compilador “agarra” em tudo, texto e instruções dadas, e transforma-os num belo documento *.pdf*. Lindo não é?

Compilador	Sistema	Site
MikTeX	Windows	http://miktex.org/
teTeX	Linux	http://www.tug.org/tetex/
OzTeX	Mac	http://www.trevorrow.com/oztex/

Table 2: Alguns compiladores de \LaTeX

3.2 Antes de começar o texto

Até agora, nas apresentações que fiz sobre \LaTeX , há uma coisa que convence qualquer um da superioridade deste sistema: a facilidade de formatação e como podemos configurá-la completamente!

3.2.1 A estrutura: classes e documentclass

O comando pelo qual todos os documentos \LaTeX começam é:

```
\documentclass{classe}1
```

É este comando que irá definir a estrutura do documento.

As classes

Se no local onde está classe colocarmos `article` obtemos uma primeira página com o aspecto seguinte:

A minha experiência em \LaTeX
Um pequeno relato
Antero Neves
anteroneves.reg@gmail.com
2 de Agosto de 2007

Conteúdo

1	Introdução	1
2	Meu primeiro contato com \LaTeX	2
3	Por onde começar	3
3.1	Pré-requisitos	3
3.2	Antes de começar o texto	4
3.2.1	A estrutura: classes e documentclass	4
4	O corpo do texto	7
4.1	Divisões do texto	8
4.2	Grandes dificuldades? . . . ou não!	9
4.3	Estrutura básica de um documento	11
5	Alguns factos mais técnicos	12
6	Conclusão	12

1 Introdução

A mudança sempre nos faz pensar e quando se faz para melhor, aí é que devemos pegar com as duas mãos! É isso que eu digo a todos que querem se aventurar neste mundo que é o \LaTeX . A verdade é que a dificuldade não está em saber \LaTeX , mas sim em ultrapassar a diferença que existe entre \LaTeX e outros.

Não podemos encarar a edição de um texto em \LaTeX da mesma maneira como encaramos no MS Word pura e simplesmente porque diferem na forma de como são feitas, pois têm abordagens diferentes, têm procedimentos diferentes e claro, têm resultados diferentes - ainda bem! :-)

Figure 1: Primeira página de um artigo

1. Todos os comandos em \LaTeX são precedidos de `\`

O comando será:

```
\documentclass{article}
```

Mas, com a simples alteração dessa palavra podemos obter uma aparência completamente diferente, e coerente ao longo do texto.

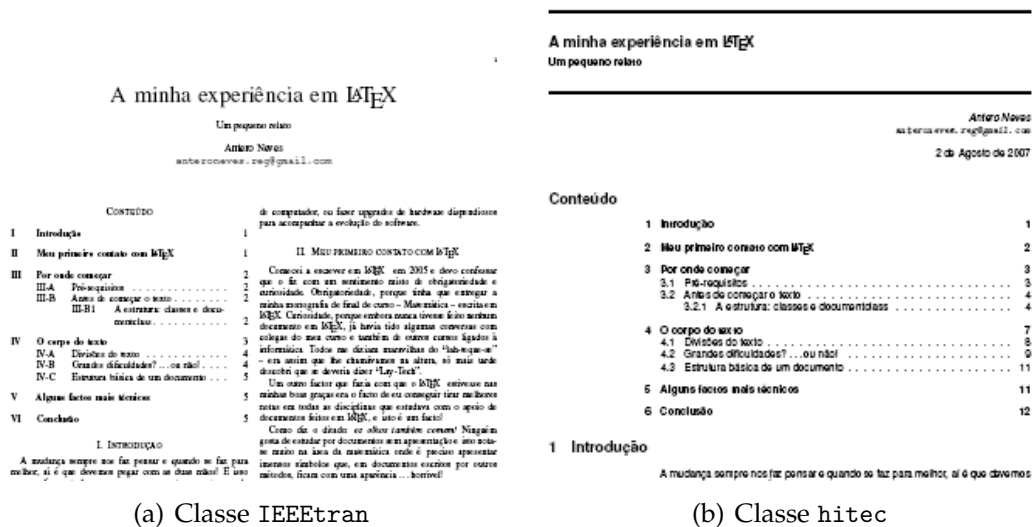


Figure 2: Mais dois layouts diferentes

Deixo aqui uma pequena lista de classes para documentos:

article	report	book
paper	amsart	amsbook
amsproc	proc	coursepaper
dtk	extarticle	hitec
IEEEtran	IEEEconf	scrartcl

Para além destas, que servem para os trabalhos “convencionais”, temos classes para apresentações: slides ou beamer(a minha favorita), classe para escrever cartas: letter, classes para escrever currículos vitae: resume ou europecv, e muitas outras que desconheço!

Cada classe apresenta formatações diferentes e requisitos diferentes, por exemplo, para se escrever uma carta temos que dar ao programa informação sobre

o remetente e o destinatário, só assim ele conseguirá formatar o documento, por outro lado não introduzimos numa carta capítulos ou secções.

O meu grande conselho no que toca a classes é experimentar e ver se ela se adapta às vossas necessidades e então usar. Nunca usar uma classe apenas pela beleza estética.

Os ficheiros que correspondem a classes, tem a extensão `cls`. No caso do exemplo apresentado: `article.cls`. As classes são, normalmente, instaladas com o MiKTeX, mas se assim não for, o ficheiro `.cls` deve estar no local onde se grava o ficheiro `.tex`. Pode também acontecer, aquando da compilação do programa, o compilador pedir para fazer download do ficheiro, e aí ele fica instalado no local adequado.

As opções

Podemos fazer alguns ajustes à classe para que ela se adapte às nossas necessidades.

A minha experiência em L ^A T _E X	
Antero Neves	
31 de Julho de 2007	
<p>Conteúdo</p> <p>1 Introdução</p> <p>2 Meu primeiro contato com L^AT_EX</p> <p>3 Por onde começar</p> <p>3.1 Pré-requisitos</p> <p>3.2 Antes de começar o texto</p> <p>3.2.1 A estrutura: classes e documentclass</p> <p>4 O corpo do texto</p> <p>4.1 Divisões do texto</p> <p>4.2 Grandes dificuldades? . . . ou não!</p> <p>4.3 Estrutura básica de um documento</p> <p>5 Alguns factos mais técnicos</p>	<p>6 Conclusão</p> <p>1 1 Introdução</p> <p>2 A mudança sempre nos faz pensar e quando se faz para melhor, aí</p> <p>3 é que devemos pegar com as duas mãos! É isso que eu digo a todos</p> <p>4 que querem se aventurar neste mundo que é o L^AT_EX. A verdade é</p> <p>5 que a dificuldade não está em saber L^AT_EX, mas sim em ultrapassar</p> <p>6 a diferença que existe entre L^AT_EX e outros.</p> <p>7 Não podemos encarar a edição de um texto em L^AT_EX da mesma</p> <p>8 maneira como encaramos no MS Word pura e simplesmente porque</p> <p>9 diferem na forma de como são feitas, pois têm abordagens diferentes,</p> <p>10 têm procedimentos diferentes e claro, têm resultados diferentes -</p> <p>11 ainda bem! >)</p> <p>12 Com este artigo, eu venho mostrar a todos que o L^AT_EX é muito</p> <p>13 simples e é muito mais fácil do que aparenta à primeira vista.</p> <p>14 Primeiro, porque os procedimentos não são assim tantos como dizem</p>

Figure 3: Artigo formatado para duas colunas e página com orientação horizontal

No exemplo acima, a classe do documento é `article`, mas são visíveis algumas alterações relativamente ao apresentado na figura 1. Para produzir os ajustamentos escrevi o seguinte:

```
\documentclass[11pt,b4paper,twocolumn,landscape]{article}
```

Outras opções que se podem colocar entre [] são:

Tamanho da letra 10pt, 11pt ou 12pt.

Tipo de papel a4paper, letterpaper, a5paper, b5paper, legalpaper, executivepaper.

Orientação de página portrait para retrato ou landscape para paisagem.

Capa titlepage para criar página de título e notitlepage para não criar.

Duas colunas twocolumn.

Equações fleqn – numeração de equações à esquerda; leqno – numeração à direita.

Impressão em dois lados twoside – impressão dos dois lados; oneside – impressão de um lado.

Primeira página dos capítulos Para obrigar a começar uma página no início de um capítulo pomos openright, se não o quisermos escrevemos openany.

A ordem pela qual se colocam as opções não interessa, o compilador interpreta-as de uma forma inteligente. Quando são omitidas as opções, a classe faz as escolhas necessárias.

3.3 Os *packages* – mais funcionalidades

Entre a instrução que define a estrutura – `\documentclass` – e o início do documento, há um rol de coisas que podemos/devemos introduzir.

Um primeiro grupo refere-se aos pacotes específicos de comandos – os *packages*. Para adicionar um determinado *package* temos que chamar usando o comando: `\usepackage{pacote}` onde no lugar de *pacote* colocamos o nome específico do *package* a incluir no processo de compilação.

Um *package* essencial para quem escreve em português é o `inputenc.sty` que com a opção `latin1` permite a inserção de texto acentuado e ç. Chamamos o *package* digitando:

```
\usepackage[latin1]{inputenc}
```

Isso facilita muito a vida dos iniciantes.

À medida que vamos precisando de mais *packages*, vamos adicionando-os no início do documento. O seu número é infundável, sempre em crescimento e existem para quase tudo, sendo o seu grande objectivo facilitar a edição de texto e a realização de tarefas mais complicadas. Os que eu uso mais regularmente, para além do já referido `inputenc` são:

babel.sty Para colocar em português todo o texto pré-definido, como “conteúdos”, “referências”, “parte”, etc. Uso o comando da seguinte forma:

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

indentfirst.sty Para que o primeiro parágrafo de cada divisão seja indentedo.

```
\usepackage{indentfirst}
```

hyperref.sty Para fazer hiperligações no documento.

```
\usepackage{hyperref}
```

graphicx.sty Para poder incluir imagens no meu texto \LaTeX .

```
\usepackage{graphicx}
```

3.4 Apresentação do trabalho e autor

A minha experiência em \LaTeX

Um pequeno relato

Antero Neves

anteroneves.reg@gmail.com

2 de Agosto de 2007

É também nesta parte que fazemos a apresentação do documento. Aí, com comandos específicos, indicamos informação sobre o autor do documento e sobre o texto em si.

Para o exemplo que se apresenta foi introduzido o seguinte código:

Figure 4: Apresentação do trabalho e autor

```
\author{Antero Neves\\
        \texttt{anteroneves.reg@gmail.com}}
\title{ A minha experiência em \LaTeX\\
        {\normalsize Um pequeno relato}}
\date{\today}
```

Os comandos essenciais são: `\author`, `\title` e `\date`.

Para que a informação apareça na primeira página do documento é necessário colocar o comando

```
\maketitle
```

no início do corpo do texto. Esta informação também é colocada automaticamente, em rodapés ou cabeçalhos, dependendo da classe.

No código encontramos várias vezes `\\`; usamos isto quando queremos obrigar a uma mudança de linha.

No comando `\date`, `\today` faz com que a data que aparece no documento seja a do dia da compilação do documento.

3.4.1 Tipo e tamanho de letra

O comando `\texttt` presente no código acima faz a transformação do tipo de letra para letra de máquina de escrever.

Exemplo:

```
\textbf{Texto a negrito}!
```

Texto a negrito!

Há outros tipos de letra que importa referir nesta fase, nomeadamente *itálico* e **negrito** que se obtêm da mesma forma mas usando `\emph{}` e `\textbf{}` respectivamente.

O tamanho do texto *Um pequeno relato* foi alterado usando o tamanho `\normalsize`. Os tamanhos de letra que se podem usar são: `\tiny`, `\scriptsize`, `\footnotesize`, `\small`, `\normalsize`, `\large`, `\Large`, `\huge`, `\Huge`.

Exemplo:

```
{\footnotesize Tamanho footnotesize}
```

Tamanho footnotesize

4 O corpo do texto

Introduzidas todas as informações podemos começar a escrever o corpo do texto.

A grande filosofia presente nesta fase da edição é: *tudo o que começa tem um fim* e é necessário dizer ao \LaTeX onde as coisas começam e onde acabam. Assim, todo o texto terá que ser escrito entre os comandos `\begin{document}` e `\end{document}`.

Exemplo:

```
\begin{document}
  Corpo do texto aqui...
\end{document}
```

É portanto fácil de entender que os comandos `\begin` e `\end` são importantíssimos na edição. São eles que definem o início e o final dos chamados *ambientes*. O ambiente de tabelas, de equação, de teorema, de inserção de imagens, de lista, etc. são definidos assim.

Outra forma de delimitar acções é usando `{` e `}`, ver o exemplo de `\textbf` acima.

4.1 Índice e divisões do texto

Para que o texto apresente um índice dos conteúdos, tal como aparece na figura 1, é necessário colocar no corpo do texto o comando: `\tableofcontents`². É normal e aconselhável que o corpo do texto apareça dividido, para melhor se compreender a sua estrutura. Em \LaTeX fazemos essas divisões através de comandos específicos muito intuitivos. São eles:

Conteúdo		
1	Introdução	1
2	Meu primeiro contato com \LaTeX	2
3	Por onde começar	3
3.1	Pré-requisitos	3
3.2	Antes de começar o texto	4
3.2.1	A estrutura: classes e documentclass	4
4	O corpo do texto	7
4.1	Divisões do texto	8
4.2	Grandes dificuldades? ... ou não!	9
4.3	Estrutura básica de um documento	11
5	Alguns factos mais técnicos	12
6	Conclusão	12

Figure 5: Tabela de conteúdos

<code>\part</code>	<code>\subsubsection</code>
<code>\chapter</code>	<code>\paragraph</code>
<code>\section</code>	<code>\subparagraph</code>
<code>\subsection</code>	

À excepção de `\part`, todos os restantes formam uma estrutura hierárquica de seccionamento.

2. Índice de imagens: `listoffigures`
Índice de tabelas `listoftables`

A grande vantagem em usar estes comando é ficarmos totalmente livres das preocupações com formatações(tipo de letra, tamanho, negrito, itálico. . .), o L^AT_EX faz tudo automaticamente.

Existem, contudo, algumas restrições ao uso de alguns tipos de secções dependendo do tipo de documento(classe do documento), assim, se o documento for do tipo book ou report, a primeira secção permitida é \chapter. Já na classe article a hierarquia tem o seu topo em \section.

Como já disse o uso destes comandos é muito intuitivo e faz-se da seguinte maneira:

```
\tipo_de_secção{Título da secção}
```

Se usarmos

```
\tipo_de_secção*{Título da secção}
```

a secção não aparece numerada. Exemplo: chapter*{A entropia}.

4.2 Tabelas

Nome	Publicações
José	17
Antónia	13
Sérgio	7

```

\begin{tabular}{|l|c|}
\hline
\textbf{Nome}& \textbf{Publicações} \\
\hline
José & 17 \\
\hline
Antónia & 13 \\
\hline
Sérgio & 7 \\
\hline
\end{tabular}

```

A inserção de tabelas pode ser feita com o ambiente tabular.sty.

Logo na primeira linha define-se o alinhamento das colunas, usando-se l – alinhamento à esquerda, c – alinhamento ao centro e r – alinhamento à direita. O símbolo | serve para fazer os limites verticais das células da tabela e também são inseridos nessa primeira linha.

Os limites horizontais são definidos pelo comando \hline

Mais uma vez o final das linhas é determinado por \\ e a mudança de colunas por &.

4.3 Listas – Items e enumerações e descrições

– Primeiro item	<code>\begin{itemize}</code>
– Segundo item	<code>\item Primeiro item</code>
– Terceiro item	<code>\item Segundo item</code>
	<code>\item Terceiro item</code>
	<code>\end{itemize}</code>
1. Primeiro item	<code>\begin{enumerate}</code>
2. Segundo item	<code>\item Primeiro item</code>
3. Terceiro item	<code>\item Segundo item</code>
	<code>\item Terceiro item</code>
	<code>\end{enumerate}</code>
<i>Látex</i> do Lat. latex, água nascente, líquido	<code>\begin{description}</code>
	<code>\item[Látex] do Lat. latex, água nascente,</code>
	<code>líquido</code>
	<code>\end{description}</code>

Para fazer listas temos as três hipóteses mencionadas acima. O seu funcionamento é simples, basta criar o ambiente que queremos com o `begin` e `end` e dentro dele identificar o começo de cada item com o comando `\item`.

4.4 Escrever matemática

Exemplo:

```
1 Se tivermos  $(\sin(\alpha) = \frac{\sqrt{3}}{2})$ 
2 sabemos que  $(\alpha = \frac{\pi}{3})$  e podemos escrever então que:
3 \begin{equation}
4   \sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}
5 \end{equation}
6
```

Se tivermos $\sin(\alpha) = \frac{\sqrt{3}}{2}$ sabemos que

$$\alpha = \frac{\pi}{3}$$

e podemos escrever então que:

$$\sin\left(\frac{\pi}{3}\right) = \frac{\sqrt{3}}{2} \quad (1)$$

A inserção de caracteres matemáticos pode ser feita de várias formas, a primeira será usando \$. Quando a inserção é feita numa linha usamos \$ para assinalar o começo e outro \$ que assinala o fim, se quisermos que a inserção seja num parágrafo à parte então usamos \$\$ para o início e \$\$ para finalizar. Hoje em dia, em vez dos \$ é usual \ (no início e \) para finalizar, e para substituir \$\$ usamos \[e \].

Há muitos outros ambientes que permitem escrever matemática, nomeadamente ambientes multi-linhas, como matrizes.

4.4.1 Ambientes multi-linhas

Para colocar expressões que ocupam mais do que uma linha, como por exemplo definir uma função por troços, podemos recorrer ao ambiente `array.sty`.

Exemplo:

```
\left[
\begin{array}{lcr}
a & b & c \\
d & e & f \\
g & h & i
\end{array}
\right]
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

Como qualquer ambiente começamos com `\begin{array}` e terminamos com `\end{array}`. Temos também as opções para alinhar o texto: `l` – esquerda, `c` – centro, `r` – direita, e definimos uma letra para cada coluna, no exemplo temos 3 colunas então escrevemos 3 letras - `lcr` - isso faz com que a primeira coluna

fique alinhada à esquerda, a segunda ao centro e a terceira à direita. Se fossem 4 colunas e as quiséssemos todas alinhadas ao centro poríamos `{cccc}`.

O funcionamento deste ambiente é muito parecido ao ambiente de tabelas.

Um aspecto importante dos ambiente matemáticos reside no tamanho dos parêntesis (curvos ou rectos) e chavetas. Para que eles tenham o tamanho correcto utiliza-se `\left` e `\right` junto aos símbolos delimitadores, estando ele à esquerda ou a direita respectivamente.

Exemplo:

```

\[(\frac{\sin \pi}{3})\]
\[\left(\frac{\sin \pi}{3}\right)\]

```

$$\left(\frac{\sin \pi}{3}\right)$$

$$\left(\frac{\sin \pi}{3}\right)$$

Temos também o ambiente `align`, mas este necessita do pacote `amsmath.sty`.

Exemplo:

```

\begin{align}
x &= y \\
a &= r \\
\nonumber 3 &= 5f \\
3 &= s
\end{align}

```

$$x = y \tag{2}$$

$$a = r \tag{3}$$

$$3 = 5f$$

$$3 = s \tag{4}$$

Este ambiente não necessita de ser introduzido num ambiente de matemático definido por `$` ou outro. Tem também a particularidade de estar definido com duas colunas e de o seu alinhamento estar pré-definido.

Todas as equações aparecem numeradas, se não o desejarmos podemos usar o comando `\nonumber` antes de cada equação, tal como se pode ver no exemplo.

4.5 Inserir imagens

No campo de tratamento de imagem, o \LaTeX é realmente muito diferente. Principalmente porque as imagens não são inseridas no ficheiro em que escrevemos, elas apenas são referidas e depois o compilador vai buscá-las. Por isso recomendo a criação de um directório no local onde criamos o ficheiro *.tex* para assim ser mais fácil ir buscar a imagem.

As imagens podem ter diversos formatos: *.jpg*, *.png*, *.eps*, *.pdf*, devem ser os mais usados.

Como já referi neste artigo, para inserir uma imagem temos que chamar o pacote `graphicx.sty`.

Exemplo:

```
\begin{figure}[h!tbp]
  \center
  \includegraphics[width=8cm]{imagens/info.pdf}
  \caption{Imagem exemplo}
  \label{Img}
\end{figure}
```

A minha experiência em \LaTeX

Um pequeno relato

Antero Neves

`anteroneves.reg@gmail.com`

2 de Agosto de 2007

Figure 6: Imagem exemplo

A imagem do exemplo é a figura 4. Como podem verificar, temos aqui outro ambiente: `figure.sty` que facilita a colocação da imagem no local pretendido, pelo uso das opções `h` – colocar exactamente onde é mencionada (o sinal `!` é um reforço desta instrução), `t` – colocar no topo, `b` – colocar no fundo e `p` – numa página só com imagens e tabelas.

O comando `\center` centra a imagem. Se quisermos alinhar à esquerda usamos `\flushleft` e para a direita `\flushright`.

A imagem em si é colocada no texto com a linha:

```
\includegraphics[width=8cm]{imagens/info.pdf}
```

Há aqui alguns pontos importantes a referir. No interior dos `[]` colocamos as medidas que a figura deve ter, podemos referir a altura – `height` ou a largura – `width`, se não colocarmos nada o tamanho será o original. Caso sejam colocados os dois, convém ter em conta a opção `keepaspectratio=true/false` que forçará, se assim for necessário a que a figura não se deforme. A linha referida ficaria assim:

```
\includegraphics[height=10cm, width=8cm, keepaspectratio=true]
{imagens/info.pdf}!
```

As linhas seguintes referem-se à informação da imagem.

`\caption` define a legenda da figura.

`\label` dá o nome à imagem para que ela possa ser referenciada ao longo do texto.

Para fazer uma referência usa-se o comando `\ref{}`.

4.5.1 Imagens inseridas em parágrafos

Exemplo:

```
\piccaption[\label{TOC}]{Tabela}
\pichskip{2cm}
\parpic[1]{\includegraphics[width=5cm]{imagens/tocs.pdf}}
Texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto
texto texto texto texto texto texto texto texto texto texto.
```

Conteúdo	
1 Introdução	1
2 Meu primeiro contato com \LaTeX	2
3 Por onde começar	3
3.1 Pré-requisitos	3
3.2 Antes de começar o texto	4
3.2.1 A estrutura: classes e documentclass	4
4 O corpo do texto	7
4.1 Divisões do texto	8
4.2 Grandes dificuldades? ... ou não!	9
4.3 Estrutura básica de um documento	11
5 Alguns factos mais técnicos	12
6 Conclusão	12

Figure 7: Tabela

Texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto
 texto texto texto texto texto texto texto texto

texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto texto.

Existe mais do que uma forma de inserir imagens em parágrafos, mas a que vou explicar aqui, e que proporciona o resultado acima, é para mim a mais completa em termos de opções, e a que produz melhores resultados.

Assim, devemos começar por chamar o *package* `picins.sty` e introduzir no preâmbulo, a linha:

```
\usepackage{picins}
```

Tal como antes, o comando `\includegraphics` coloca a imagem no documento, e as suas opções são exactamente as mesmas. Mas agora essa instrução está inserida numa outra: `\parpic`.

No exemplo dado temos a linha:

```
\parpic [l]{\includegraphics [width=3.5cm]{imagens/tocs.pdf}}
```

podemos ver que o comando `\parpic` vem acompanhado com uma opção `[l]`, esta opção faz com que a imagem fique na parte esquerda do parágrafo, mas há muitas outras opções que acompanham este comando.

A instrução completa é:

```
\parpic(lar., alt.)(x,y)[opções][posição]{imagem}
```

E as opções são:

- `lar.` e `alt.` definem a largura e altura da caixa que rodeia a imagem.
- `x` e `y` marcam o deslocamento da figura relativamente à moldura.
- Na parte seguinte podemos colocar duas opções, uma de cada grupo que se segue:
 1º grupo

- l coloca a imagem na parte esquerda do parágrafo.
- r coloca a imagem na parte direita do parágrafo.

2º Grupo

- f - para inserir uma moldura
 - d - inserir uma moldura a tracejado
 - o - moldura oval
 - s - moldura com sombra
 - x - moldura a 3D
- Na parte da posição tratamos do alinhamento da figura na moldura com as tradicionais opções:
- l - esquerda
 - t - topo
 - r - direita
 - b - fundo

Antes da linha `\parpic` podemos inserir vários comandos que influenciam a aparência do documento:

- `\picskip{n}`, onde `n` define o número de linhas indentadas. Por defeito elas são indentadas até ao final da imagem.
- `\pichskip{distância}`, marca a distância horizontal do texto à imagem.
- `\piccaption[] {legenda}`, entre `[]` podemos inserir algumas opções como o nome que o programa dá à imagem usando `\label{nome}`, em `legenda` colocamos a legenda a colocar.

4.5.2 Múltiplas imagens

Há também a possibilidade de colocar imagens lado a lado. Para isso chamamos mais um pacote: `subfigure.sty`.

Exemplo:

```
\begin{figure}[h!]  
  \center  
  \subfigure[a][Subcaption1]{\includegraphics[width=4cm]  
  {imagens/tocs.pdf}}  
  \qqquad  
  \subfigure[b][Subcaption2]{\includegraphics[width=4cm]  
  {imagens/tocs.pdf}}  
  \qqquad  
  \subfigure[c][Subcaption1]{\includegraphics[width=4cm]  
  {imagens/tocs.pdf}}  
  \caption{Imagens lado a lado}  
\end{figure}
```

Conteúdo		Conteúdo		Conteúdo	
1 Introdução	1	1 Introdução	1	1 Introdução	1
2 Meu primeiro contato com \LaTeX	2	2 Meu primeiro contato com \LaTeX	2	2 Meu primeiro contato com \LaTeX	2
3 Por onde começar	3	3 Por onde começar	3	3 Por onde começar	3
3.1 Pré-requisitos	3	3.1 Pré-requisitos	3	3.1 Pré-requisitos	3
3.2 Antes de começar o texto	4	3.2 Antes de começar o texto	4	3.2 Antes de começar o texto	4
3.2.1 A estrutura: classes e documentclass	4	3.2.1 A estrutura: classes e documentclass	4	3.2.1 A estrutura: classes e documentclass	4
4 O corpo do texto	7	4 O corpo do texto	7	4 O corpo do texto	7
4.1 Divisão de texto	8	4.1 Divisão de texto	8	4.1 Divisão de texto	8
4.2 Grandes dificuldades? ... ou não!	9	4.2 Grandes dificuldades? ... ou não!	9	4.2 Grandes dificuldades? ... ou não!	9
4.3 Estrutura básica de um documento	11	4.3 Estrutura básica de um documento	11	4.3 Estrutura básica de um documento	11
5 Alguns factos mais técnicos	12	5 Alguns factos mais técnicos	12	5 Alguns factos mais técnicos	12
6 Conclusão	12	6 Conclusão	12	6 Conclusão	12

(a) Subcaption1

(b) Subcaption2

(c) Subcaption1

Figure 8: Imagens lado a lado

Então, para começar, colocamos no preâmbulo a linha:

```
\usepackage{subfigure}
```

Usamos depois o comando `\subfigure` com as opções devidas, mas dentro do ambiente `\figure`.

A instrução completa é:

```
\subfigure[ref1][Legenda1]{\includegraphics[width=5cm]{caminhoimagem1}}
```

A explicação deste comando é simples.

- `ref1` - nome que a subfigura terá no documento.
- `Legenda1` - legenda que a subfigura terá no documento.

- por fim usamos o comando `\includegraphics` como foi explicado anteriormente.

As opções `ref1` e `Legenda1` são facultativas.

Por último coloca-se a legenda geral usando:

```
\caption{Imagens lado a lado}
```

também esta linha é facultativa.

O comando `\qqquad`, usado no exemplo, controla o espaçamento entre as imagens. Podemos recorrer a outros comandos para fazer outro tipo de espaços, como `\quad` ou `\mbox{}`.

5 Alguns factos mais técnicos

Muitos de nós têm ferramentas de edição de texto que já vêm instaladas quando compramos um computador, o tempo avança e muitas vezes é necessário actualizar o software, a verdade é que essas actualizações ficam caras, e muitas vezes não é só o preço do programa em si, mas também do hardware porque os programas exigem mais espaço, mais capacidade de processamento e mais RAM. A RAM necessária para algumas ferramentas, ou pelo menos para activar todos os seus recursos, atinge hoje 1GB!!! E o processador só é mais usado pelo \LaTeX aquando da compilação, enquanto que as outras ferramentas necessitam dele continuamente.

Há uns anos atrás fazíamos os mesmos trabalhos em computadores com um terço das capacidades dos computadores de hoje. E eu deixo-vos apenas uma questão: afinal em que é que os trabalhos actuais são melhores do que os que se faziam há uns anos atrás?

6 Conclusão

Com a crescente exigência por parte de professores, colegas de trabalho e da sociedade geral, torna-se essencial uma produção de textos científicos de qualidade. O \LaTeX é um enorme contributo para a melhoria dessa qualidade relativamente a ferramentas de edição de texto “tradicionais” em especial na área da Matemática.

Parece-me que a maioria das pessoas que vê um documento concebido em \LaTeX pensa imediatamente que ele está bem feito e que fazer uma coisa igual deve ser muito difícil. Nada mais errado! Trabalhar com \LaTeX não é difícil, é sim diferente daquilo a que fomos habituados, e essa diferença faz com que seja fundamental a compreensão do mecanismo e da filosofia reinantes. Neste artigo procurou-se explicar de forma simples estes processos.

Tentou-se também facilitar o começo de alguns curiosos e iniciantes do mundo \LaTeX pela apresentação de alguns procedimentos básicos, algumas opções nem sempre óbvias e pela apresentação da estrutura de um documento \LaTeX .

Espero ter atingido esses objectivos e que mais gente se renda definitivamente ao \LaTeX depois de ler estas linhas.