



**Créditos** Estes guiões constituem um trabalho continuado de refinamento que contou com a colaboração de João Paulo Barraca, Diogo Gomes, André Zúquete, João Manuel Rodrigues, António Adrego da Rocha, Tomás Oliveira e Silva, Sílvia Rodrigues e Óscar Pereira.

TEMA 3

# Produção de documentos com **L<sup>A</sup>T<sub>E</sub>X**

## Objetivos:

- Compilação de documentos L<sup>A</sup>T<sub>E</sub>X.
- Estruturação de documentos L<sup>A</sup>T<sub>E</sub>X.
- Referências bibliográficas com L<sup>A</sup>T<sub>E</sub>X e BibTeX.

## 3.1 Introdução

O  $\text{\LaTeX}$  é uma ferramenta de apoio à produção de documentos complexos (particularmente de índole técnica). O seu paradigma de funcionamento é completamente diferente do de outras ferramentas mais populares atualmente, como o *Word* da *Microsoft*. Enquanto editores tradicionais, denominados What You See Is What You Get (WYSIWYG) focam-se na construção simultânea de todo o documento, incluindo aspeto gráfico e conteúdo, o  $\text{\LaTeX}$  pressupõe que o autor se deve focar no conteúdo, categorizando apenas o texto produzido, sem se preocupar imediatamente com o aspeto. Este tipo de sistemas são denominados por What You See Is What You Mean (WYSIWYM).

Portanto, ao escrever um documento utilizando  $\text{\LaTeX}$ , devemos focar-nos em adicionar conteúdo e em especificar o que queremos como título, secção, tabela, ou que imagens queremos incluir. Aspetos como o estilo, tipo de letra, localização das imagens, espaçamentos, paginação e mesmo hifenação ficam a cargo do  $\text{\LaTeX}$ . O autor pode refinar estes pontos, mas à partida não terá de se preocupar com eles. (Para mais informação sobre como os modificar, pode consultar [1][2].)

O  $\text{\LaTeX}$  é a ferramenta por excelência para a realização de relatórios de carácter técnico, publicações científicas e mesmo guiões para aulas (de que este é um exemplo). Neste guião iremos descrever os conceitos elementares para produzir um documento minimamente estruturado em  $\text{\LaTeX}$ .

## 3.2 Ações de preparação

Esta aula irá utilizar programas que estão disponíveis nos computadores da sala. Se realizar este guião durante a aula, não é necessário qualquer ação adicional.

Se realizar este passo em casa, poderá instalar todas as ferramentas adicionais na imagem virtual fornecida através do comando:

```
apt-get install texmaker
```

Se quiser instalar o  $\text{\LaTeX}$  no seu computador, e dependendo do sistema operativo que utilize, instale o TexLive ou o MikTeX.

Se estiver a utilizar um sistema *Linux* como o Ubuntu, pode executar:

```
apt-get install texlive
```

### 3.3 Compilação de documentos $\text{\LaTeX}$

Os documentos  $\text{\LaTeX}$  são produzidos editando ficheiros de texto com um editor elementar, como o **vim**. Esta maneira de trabalhar facilita a edição colaborativa pois reduz problemas de compatibilidade entre sistemas. Qualquer editor de texto pode ser utilizado, o que permite que um documento seja facilmente editado. Este mesmo guião foi parcialmente realizado em sistemas *Windows*, *OS X* e *Linux*.

Os ficheiros  $\text{\LaTeX}$  são então facilmente alterados por humanos, contendo o texto que deverá constar do documento final a produzir, juntamente com comandos relativas à sua formatação.

De notar que ao editar um documento através do sistema  $\text{\LaTeX}$ , não estaremos a ver como o documento realmente irá ser apresentado. Para isso é necessário um passo de compilação. Esta compilação converte o documento  $\text{\LaTeX}$  para um outro formato, sendo que o mais utilizado é o Portable Document Format (PDF). Pode considerar que o conceito é semelhante ao da compilação de um programa *Java* no formato **.java** para um ficheiro **.class**.

Existem editores especializados que permitem ter uma vista quase em tempo real do documento final. Na realidade, estes editores estão constantemente a compilar o documento, apresentando depois o resultado ao utilizador. Um exemplo é o **gummi**, outro o **texmaker**, mas existem ainda muitos outros<sup>1</sup>.

Para compilar um documento  $\text{\LaTeX}$  é necessário executar o compilador. Num interpretador de linhas de comandos, como a **bash**, isso é feito colocando a seguir ao nome do compilador (por exemplo, **pdflatex**) o nome do ficheiro a compilar (a extensão **.tex** pode ser omitida). Por exemplo, pressupondo que o nosso ficheiro se chama **hello.tex**, poderemos compilar o ficheiro para PDF através do comando:

```
$ pdflatex hello.tex
```

O resultado deverá ser um conjunto de ficheiros auxiliares (que podem ser removidos), e um documento com extensão PDF. Na Secção 3.12 voltaremos a este assunto.

---

<sup>1</sup>Veja <http://tex.stackexchange.com/questions/339/latex-editors-ides>

### Exercício 3.1

No sítio da disciplina existe um ficheiro chamado **hello.tex**. Obtenha este ficheiro. O conteúdo deste ficheiro ainda não é relevante. Compile este ficheiro e verifique que é produzido um ficheiro com extensão PDF.

Verifique que consegue visualizar o conteúdo do ficheiro  $\text{\LaTeX}$  utilizando o comando **vim**.

Pode igualmente editar o ficheiro através da aplicação **texmaker**. A vantagem desta aplicação é já ter algumas funcionalidades que auxiliam a produção de documentos  $\text{\LaTeX}$ .

Para visualizar o documento produzido, pode utilizar as aplicações **epdfview** ou **evince**.

## 3.4 Estrutura obrigatória de um documento

A estrutura obrigatória de um documento passa em primeiro lugar, e obrigatoriamente, pela definição do seu tipo de estrutura, pela definição do tipo de letra base (tipo e tamanho), pela definição da dimensão da página, pela definição de como cada página está organizada (uma coluna, duas colunas), etc.

Esta estrutura é definida pelo comando `\documentclass[options]{class}`. O seu parâmetro obrigatório é a classe de documento, a qual define o tipo base de documento. Eis alguns exemplos de classes: **book**, **report**, **article**. Os parâmetros opcionais podem ser (lista não exaustiva):

**a4paper**, **a5paper**, etc. : dimensão da folha.

**10pt**, **11pt**, **12pt** : dimensão do tamanho da letra.

**onecolumn** ou **twocolumn** : texto com uma ou duas colunas.

**oneside** ou **twoside** : só frente ou frente e verso.

Este é um exemplo para especificar um relatório em páginas a4, com letras de 11pt, a duas colunas e frente-e-verso:

```
\documentclass[a4paper,11pt,twocolumn,twoside]{report}
```

Após esta especificação, o conteúdo efetivo do documento deverá ser colocado entre dois comandos: `\begin{document}` e `\end{document}`.

---

```
\documentclass[a4paper]{article}

\begin{document}

% conteúdo do documento

\end{document}
```

---

### Exercício 3.2

Copie o documento `hello.tex` para um novo ficheiro e produza o PDF respetivo. Coloque novas opções e/ou altere as que já existem. Observe o resultado. Coloque muito mais conteúdo, de forma a produzir múltiplas páginas, altere novamente as opções, e observe o resultado.

## 3.5 Carateres especiais do $\text{\LaTeX}$

No ficheiro que já compilou pode verificar que, além do texto, existem diversas instruções, que denominámos por comandos. Estes usam carateres que normalmente não fazem parte do conteúdo de um texto de um documento, a saber: `\`, `{`, `}`, `[`, `]`, `$`, `%`, `~`, `#`, `_`, `^`, e `&`. Neste momento é prematuro introduzir o significado de todos eles, vamos apenas apresentar alguns já de seguida.

### 3.5.1 O carácter `'\'`

O carácter `'\'` é o símbolo base da definição de comandos, daí que normalmente está no seu início:

`\o`  $\rightarrow$   $\emptyset$

`\oe`  $\rightarrow$   $\text{\oe}$

`\LaTeX`  $\rightarrow$   $\text{\LaTeX}$

Para escrever o carácter `'\'` num texto (o que é raro) é preciso recorrer a um comando:

`\textbackslash` → \

### Exercício 3.3

Edite o documento e adicione estes comandos depois do texto “Hello World!”. Não se esqueça de compilar o documento e verificar o resultado!

#### 3.5.2 O caráter ‘\$’

O caráter ‘\$’ é usado para sinalizar a entrada e saída do modo matemático. Por omissão o conteúdo fonte de um documento  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  é interpretado em modo de texto, entrando no modo matemático e saindo do mesmo com este caráter. O modo matemático será abordado mais adiante.

`text mode, $math mode$, text mode` → text mode, *mathmode*, text mode

Para escrever o caráter ‘\$’ num texto é preciso recorrer a um comando:

`\$` → \$

#### 3.5.3 Os caracteres ‘{’ e ‘}’

Estes caracteres, chavetas, servem para dois fins: definir um contexto e definir um parâmetro de um comando.

Um contexto consiste num bloco de texto, entre as duas chavetas (a de abrir e a de fechar), que delimita o universo de abrangência de comandos de formatação expressos nesse mesmo bloco. Ou seja, uma formação imposta dentro de um bloco não tem efeito fora do bloco. Neste âmbito, as chavetas servem também para delimitar corretamente os comandos que não possuem parâmetros, nomeadamente para não interpretar o espaço que é usado após os mesmos como sendo um terminador.

`\o e` → ø e (o espaço é interpretado como terminador de ‘\o’)

`{\o} e` → ø e (o espaço é preservado como separador de ‘\o’ e ‘e’)

Quando os comandos possuem parâmetros obrigatórios, estes são normalmente indicados entre chavetas (alguns só funcionam em modo matemático, delimitados por \$) :

`\textbf{a}` → **a**

`\vec{x}` →  $\vec{x}$

`\sqrt{x}` →  $\sqrt{x}$

`\textsc{texto em Small Caps}` → TEXTO EM SMALL CAPS

`\fbox{texto emoldurado}` → texto emoldurado

`\footnote{Nota de rodap'e}` → <sup>2</sup>

Quando os comandos possuem mais do que um parâmetro, cada um dos parâmetros é indicado entre chavetas:

`\frac{x}{y}` →  $\frac{x}{y}$

Para escrever os caracteres '{' ou '}' num texto (o que é raro) é preciso recorrer a um comando:

`\{ \}` → { }

### Exercício 3.4

Edite o documento e adicione alguns destes comandos junto do restante texto.

Pode igualmente experimentar outras formatações de texto como **negrito** (`\textbf{negrito}`), *itálico* (`\textit{itálico}`), ou como uma máquina de escrever (`\texttt{máquina de escrever}`).

---

<sup>2</sup>Nota de rodapé

### 3.5.4 Os caracteres '[' e ']'

Estes caracteres normalmente são interpretados literalmente (i.e., sem qualquer significado especial) exceto quando delimitam opções de comandos:

$$\boxed{\backslash\mathrm{sqrt}[3]{x}} \rightarrow \sqrt[3]{x}$$

### 3.5.5 O carácter '%'

O carácter '%' é usado para sinalizar o início de texto que não deverá ser interpretado (usualmente um comentário), até ao final da linha corrente.

$$\boxed{\text{texto interpretado, \%texto ignorado}} \rightarrow \text{texto interpretado,}$$

Para escrever o carácter '%' num texto é preciso recorrer a um comando:

$$\boxed{\backslash\%} \rightarrow \%$$

### 3.5.6 O carácter '~'

O carácter '~' é usado para representar um espaço entre dois elementos que nunca devem ficar separados em duas linhas consecutivas. Por exemplo, quando se escreve “*a arquitetura apresentada na figura 5*” não é correto que ocorra uma translineação (mudança de linha) entre “*figura*” e “*5*”, porque isso prejudica a leitura do texto. Para sinalizar esse facto, escreve-se:

$$\boxed{\text{a arquitetura apresentada na figura}\sim\text{5}} \rightarrow \text{a arquitetura apresentada na figura 5}$$

Para realmente escrever o carácter '~' num texto é preciso recorrer a um comando:

$$\boxed{\backslash\text{textasciitilde}} \rightarrow \sim$$

## 3.6 Funcionalidades adicionais (linguísticas)

Normalmente nos documentos  $\text{\LaTeX}$  usam-se funcionalidades para além das incluídas da especificação base do compilador. Essas funcionalidades são disponibilizadas em pacotes (*packages*), os quais são normalmente indicados no início de um documento através do comando `\usepackage[options]{packages}[version]`. É também comum usar um comando destes por cada pacote usado.



Um exemplo de pacote que lhe poderá ser útil é o que permite interpretar corretamente caracteres para além dos contemplados na tabela American Standard Code for Information Interchange (ASCII)[3]<sup>3</sup>, de que são exemplo todos os caracteres latinos acentuados e os caracteres de outros alfabetos que não o latino (grego, cirílico, árabe, etc.). Este pacote é o **inputenc**, que tem como opção a escolha do modo como os caracteres não ASCII estão codificados no document fonte:

```
\usepackage[encoding name]{inputenc}
```

Este guião, por exemplo, foi escrito em L<sup>A</sup>T<sub>E</sub>X com codificação utf8<sup>4</sup>, o que implicou a inclusão deste pacote do seguinte modo:

```
\usepackage[utf8]{inputenc}
```

### Exercício 3.5

Observe o resultado do PDF obtido quando se usa ou não o pacote (experimente comentar a linha da sua inclusão).

Outro pacote que é útil para português é o **babel**, que ensina o L<sup>A</sup>T<sub>E</sub>X a realizar corretamente a translineação de palavras:

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

Caso um documento inclua textos de várias línguas, como português e inglês, tal é suportado pelo pacote:

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

Finalmente, outro pacote que é útil para português é o **fontenc**, que indica o tipo de codificação de caracteres do resultado da compilação do L<sup>A</sup>T<sub>E</sub>X. Aconselha-se o uso da codificação T1, porque a mesma permite lidar convenientemente com caracteres acentuados.

```
\usepackage[T1]{fontenc}
```

---

<sup>3</sup>Também pode consultar: <http://en.wikipedia.org/wiki/ascii>

<sup>4</sup>utf8 é uma codificação que pretende ser uniforme para todas as línguas.

### Exercício 3.6

Crie um novo ficheiro contendo texto em português e em inglês. Cada parágrafo deverá ocupar mais do que uma linha de texto e deverá ter uma palavra translineada.

Verifique que o  $\text{\LaTeX}$  procede à translineação automática do texto apresentado. No entanto esta poderá não ser a mais adequada à linguagem em causa.

O comando `\selectlanguage{language}` permite especificar qual a língua do texto que se lhe segue. Use-o.

## 3.7 Dimensão das letras

A dimensão base das letras é definida com o comando `\documentclass`, como vimos na Secção 3.4. Este tamanho é indicado com o comando `\normalsize`. relativamente a este tamanho o  $\text{\LaTeX}$  possui outras dimensões, a saber:

Comando	Resultado para este guião
<code>\tiny</code>	texto de exemplo
<code>\scriptsize</code>	texto de exemplo
<code>\footnotesize</code>	texto de exemplo
<code>\small</code>	texto de exemplo
<code>\normalsize</code>	texto de exemplo
<code>\large</code>	texto de exemplo
<code>\Large</code>	texto de exemplo
<code>\LARGE</code>	texto de exemplo
<code>\huge</code>	texto de exemplo
<code>\Huge</code>	texto de exemplo

Estas indicações da dimensão do tipo de letra pode ser utilizando ao longo do texto, de forma a alterar palavras ou mesmo letras individuais. No entanto, não é comum de ser utilizado desta forma pois irá produzir um documento com pouca coerência.

---

Posso `{\small}` mudar de `{\huge}` tamanho de letra sempre que me apetecer. Essa `\Large` mudança permanece em efeito `{\normalsize}` até ao próximo comando de mudança de tamanho, caso exista, `\normalsize\space` ou até ao fim do bloco (delimitado por `\{` e `\}`) onde está inserida.

---



Posso mudar de tamanho de letra sempre que me apetecer. Essa mudança permanece em efeito até ao próximo comando de mudança de tamanho, caso exista, ou até ao fim do bloco (delimitado por `{` e `}`) onde está inserida.

Como pode reparar, quando se escreve um documento, não existe a necessidade de ajustar o tamanho das letras para valores específicos. Apenas se indica o que se pretende que seja maior ou menor, com base na referência do documento. Desta forma ele ficará sempre coerente.

## 3.8 Elementos estruturais de documentos

### 3.8.1 Título

Nos documentos com título o mesmo é criado com recurso aos seguintes comandos:

`\title{título}`: Define o texto do título. Se o mesmo tiver de ser dividido por várias linhas deverá ser usado o comando `\\` para mudar de linha.

`\author{autores}`: Define os autores. Também neste caso os autores podem ser divididos por várias linhas usando o comando `\\` para mudar de linha. Nesta parte podem ainda ser incluídos outros atributos relativos aos autores para além do seu nome, como afiliações, endereços, etc.

`\date{data}`: Define a data de criação do documento, que é apresentada no título. O comando `\today` fornece a data atual. Para não incluir data deve-se usar uma data vazia (ou seja, nada entre as chavetas).

`\maketitle`: Cria o título de acordo com o texto do mesmo, os autores e a data antes definidos.

Convém que estes comandos sejam colocados antes do início formal do documento, isto é, antes de `\begin{document}`, tal como acontece no ficheiro **hello.tex**.

### 3.8.2 Partes, capítulos e secções

Os documentos podem ser seccionados com os seguintes comandos, cujo nome é sugestivo:

`\part{title}`: inicia uma parte de documento (pode englobar vários capítulos);

`\chapter{title}`: inicia um capítulo (pode englobar vários secções);  
`\section{title}`: inicia uma secção (pode englobar vários subsecções);  
`\subsection{title}`: inicia uma subsecção (pode englobar vários sub- subsecções);  
`\subsubsection{title}`: inicia uma sub-subsecção (pode englobar vários parágrafos);

Todos estes comandos de seccionamento têm obrigatoriamente um título, o qual terá a fonte e dimensão de letra apropriadas escolhidas automaticamente pelo  $\text{\LaTeX}$ . Eles são igualmente numerados de forma automática e segundo estilos definidos pelo tipo de documento (numeração romana ou árabe, letras, etc.). Mais uma vez, ao utilizar  $\text{\LaTeX}$ , o autor especifica qual a função do texto, deixando a composição gráfica a cargo do  $\text{\LaTeX}$ .

### Exercício 3.7

Experimente usar todos estes comandos de seccionamento num documento. Note que nem todos são possíveis em cada tipo de documento (v.g. com documentos do tipo **article** não existe `\chapter`).

### Exercício 3.8

Experimente incluir o comando `\tableofcontents` logo no início do texto útil do documento. Observe o resultado após compilar o documento. Altere alguns dos títulos das secções, compile e observe o resultado. Poderá ter de repetir a compilação para a criação correta deste índice.

Alguns dos títulos criados por estes comandos, como os de `\part`, `\chapter` e `\tableofcontents`, incluem uma parte em inglês. Esta parte é automaticamente colocada pelo  $\text{\LaTeX}$  de acordo com variáveis pré-definidas e com a língua definida, as quais podem ser alteradas no documento com os seguintes comandos:

```
\renewcommand{\partname}{Parte}  
\renewcommand{\chaptername}{Tema}  
\renewcommand{\contentsname}{Índice}
```

Não é possível alterar o nome das seções pois uma seção apenas inclui um número e um título. Pelo contrário, um capítulo inclui um texto (ex., Capítulo).

Se for utilizado o package `hyperref` é possível definir os nomes que são inseridos no texto a quando de referências para outras partes do documento (ex. Como explicado na Seção 1.1), utilizando o comando `\autoref`. Neste caso seria utilizado:

---

```
\usepackage{hyperref}
```

```
....  
\renewcommand{\sectionname}{Seção}  
..
```

```
\section{Teste}  
\label{sec:teste}
```

Como foi apresentado na `\autoref{sec:teste}`...

---

### Exercício 3.9

Altere a ordem de declaração das línguas junto do pacote **babel**. Volte a compilar e verifique quais os títulos fornecidos.

Altere novamente o seu documento para incluir estas alterações logo no início do texto útil do documento. Observe o resultado após compilar o documento.

### 3.8.3 Listas de itens

O  $\text{\LaTeX}$  inclui várias formas de listas de itens, tanto numeradas como não numeradas. Estas listas são automaticamente indentadas pelo  $\text{\LaTeX}$ .

Uma lista não numerada é criada através do ambiente **itemize**:

---

```
Uma lista não numerada é:  
\begin{itemize}  
  \item este é um item;  
  \item este é outro item;  
  \item etc.  
\end{itemize}  
Aqui retoma a indentação.
```

---



Uma lista não numerada é:

- este é um item;
- este é outro item;
- etc.

Aqui retoma a indentação.

onde cada item é iniciado através do comando `\item`. Em cada comando `\item` pode indicar um conteúdo alternativo para o símbolo que o  $\text{\LaTeX}$  usa para iniciar o item (v.g. `\item[+]` usa o

símbolo '+' , ver abaixo):

---

```
Uma lista não numerada é:  
\begin{itemize}  
  \item[+] este é um item;  
  \item[+] este é outro item;  
  \item[+] etc.  
\end{itemize}  
Aqui retoma a indentação.
```

---



Uma lista não numerada é:

- + este é um item;
- + este é outro item;
- + etc.

Aqui retoma a indentação.

Outra forma de criar uma lista não numerada é através do ambiente **description**, onde cada item é iniciado por texto (ou título) arbitrário em vez de um símbolo gráfico:

---

```
Uma lista não numerada é:  
\begin{description}  
  \item[UA] - Univ. de Aveiro;  
  \item[UC] - Univ. de Coimbra;  
  \item[UP] - Univ. do Porto.  
\end{description}  
Aqui retoma a indentação.
```

---



Uma lista não numerada é:

- UA** - Univ. de Aveiro;
- UC** - Univ. de Coimbra;
- UP** - Univ. do Porto.

Aqui retoma a indentação.

onde cada item é iniciado através do comando `\item[title]`; o parâmetro opcional (que pode ser omitido) é escrito em negrito.

As listas numeradas são criadas através do ambiente **enumerate**:

---

```
Uma lista numerada é:  
\begin{enumerate}  
  \item este é um item;  
  \item este é outro item;  
  \item etc.  
\end{enumerate}  
Aqui retoma a indentação.
```

---



Uma lista numerada é:

1. este é um item;
2. este é outro item;
3. etc.

Aqui retoma a indentação.

onde cada item é iniciado através do comando `\item` e numerado automaticamente pelo  $\text{\LaTeX}$ .

A numeração segue um formato por omissão (v.g. numeração árabe) a qual pode ser mudada em cada lista logo após a sua iniciação, como neste exemplo:

---

```
Uma lista numerada é:  
\renewcommand{\theenumi}{\Roman{enumi}}  
\begin{enumerate}  
  \item Este é um item;  
  \item Este é outro item;  
  \item etc.  
\end{enumerate}  
Aqui retoma a indentação.
```

---



Uma lista numerada é:

- I. Este é um item;
- II. Este é outro item;
- III. etc.

Aqui retoma a indentação.

onde cada item é numerado com numeração romana maiúscula; o termo **enumi** representa o contador usado na listagem. No exemplo abaixo mostra-se como se pode misturar vários tipos de numeração:

---

```
Múltiplas listas numeradas e encadeadas:
\renewcommand{\theenumi}{\Roman{enumi}}
\begin{enumerate}
\item Este é um item, com subitems:
  \renewcommand{\theenumi}{\alpha{enumi}}
  \begin{enumerate}
\item Este é um item;
\item Este é outro item;
\item etc.
  \end{enumerate}
\item Este é outro item;
\item etc.
\end{enumerate}
Aqui retoma a indentação.
```

---



Múltiplas listas numeradas e encadeadas:

I. Este é um item, com subitems:

- a) Este é um item;
- b) Este é outro item;
- c) etc.

II. Este é outro item;

III. etc.

Aqui retoma a indentação.

No primeiro nível usou-se numeração Romana maiúscula; no segundo nível usaram-se letras minúsculas (começando em 'a'). **Nota:** a indentação usada neste último exemplo é irrelevante para o  $\text{\LaTeX}$ , ela apenas facilita a compreensão do leitor.

### 3.8.4 Objetos flutuantes: figuras e tabelas

No  $\text{\LaTeX}$  existe um conceito, denominado de objetos flutuantes (*floats*), que consiste na existência de objetos cujo lugar não precisa de ser fixo, porque pode ser referenciado de outra forma (v.g. através de um número de referência). Há dois tipos de objetos flutuantes: figuras e tabelas.

As figuras e as tabelas são em tudo idênticas, exceto na maneira como é apresentada a sua legenda. São objetos que podem conter qualquer conteúdo (texto, imagens, etc.) e que possuem pelo menos



uma legenda. As legendas são numeradas e o seu número pode ser usado noutros locais do texto para referir o objeto.

Uma figura é declarada do seguinte modo:

---

```
\begin{figure}[h]
\centerline{\fbox{Conteúdo da figura, pode ser texto, imagem, etc.}}
\caption{Legenda da figura}
\end{figure}
```

---

e o resultado é a figura abaixo:

Conteúdo da figura, pode ser texto, imagem, etc.
--

Figura 3.1: Legenda da figura

A opção 'h' indicada no início do ambiente **figure** indica que a figura deverá primordialmente ser colocada no lugar onde aparece em termos de fluxo de texto. Mas tal nem sempre é possível. O  $\text{\LaTeX}$  possui heurísticas internas que determinam a melhor localização de um objeto flutuante de forma a que o documento fique mais equilibrado. O resultado é que os objectos flutuantes nem sempre irão ficar onde são declarados de forma a melhorar o aspeto final do documento.

Outras opções para além da 'h' são:

- t** - A figura é colocada no topo da página.
- b** - A figura é colocada no fundo da página.
- p** - A figura é colocada numa página isolada.

Várias destas opções podem ser usadas, indicando uma ordem de preferência para o  $\text{\LaTeX}$ .

Tanto as figuras como as tabelas podem ser referidas através do seu número. Em  $\text{\LaTeX}$  essa funcionalidade é obtida através dos comandos `\label` e `\ref`. Se utilizar o *package hyperref*, é possível também usar o comando `\autoref`. Estes comandos são descritos em maior pormenor na secção seguinte.

As tabelas seguem exatamente as mesmas regras que as figuras, simplesmente usa-se o ambiente **table** e a legenda é identificada de forma diferente. Assim, uma tabela é declarada do seguinte modo:

---

```
\begin{table}[htp]
\caption{Exemplo de uma tabela}
\centerline{Conteúdo de uma tabela}
\label{tabela-exemplo}
\end{table}%
```

---

e o resultado é a tabela abaixo:

Tabela 3.1: Exemplo de uma tabela

Conteúdo de uma tabela

Note-se que nesta tabela usou-se o comando `\label` para a referir neste texto pelo seu número (Tabela 3.1, obtido através da etiqueta `tabela-exemplo` no comando `\autoref{tabela-exemplo}`). Veja a secção seguinte para obter mais pormenores sobre este tipo de comandos. Também nesta tabela usou-se o comando `\caption` antes do conteúdo da tabela, o que causou que a legenda da tabela aparecesse no seu topo.

### Exercício 3.10

Edite o seu documento e coloque-lhe várias figuras e tabelas, contendo uma ou mais figuras/tabelas por objeto flutuante. Experimente as diversas colocações e veja o que acontece com cada uma delas. Confirme também que os objetos flutuantes do mesmo tipo (figuras ou tabelas) nunca ficam por uma ordem diferente daquela em que são declaradas, muito embora possam ficar fora do local onde foram indicadas no documento fonte.

### 3.8.5 Referências a partes do texto

Qualquer identificador numérico usado num documento `LATEX`, seja ele de parte, capítulo, secção, lista numerada, figura, tabela, e outros, pode ser usado no texto através dos comandos `\label`, `\ref` e `\autoref`.

O comando `\label` associa uma etiqueta a um número usado num dos elementos acima referidos (parte, capítulo, etc.). Para isso, o normal é colocar a etiqueta imediatamente após o comando que produz o número que se pretende, muito embora isso não seja estritamente necessário. Por exemplo, o início desta secção foi escrito com o seguinte conteúdo:

```
\subsection{Referências a partes do texto}
\label{refs.Section}
```

Qualquer identificador numérico usado num documento `{LATEX}`, seja ele de parte, capítulo, secção, lista numerada, figura, tabela, e outros, pode ser usado no texto através dos comandos ...

Assim, o comando `\ref{refs.Section}` produzirá o número 3.8.5. A vantagem deste método é que o texto pode ser livremente alterado, inclusive alterada a sua ordem, que os números utilizados para as imagens, figuras e as respetivas referências é mantido de forma ordenada. Também pode ser usado o método `\autoref{refs.Section}` que irá gerar uma descrição mais longa: Subsecção 3.8.5.

O uso destes comandos obriga normalmente a duas compilações de um documento para que os valores apresentados pelas referências estejam corretos. Na primeira compilação é obtido e guardado o número associado a cada etiqueta, na segunda compilação o número associado a cada etiqueta é usado onde quer que a etiqueta seja referida. Deste modo, é perfeitamente possível ter uma referência a uma etiqueta que aparece mais tarde no texto.

Mais uma vez, o autor não tem de se preocupar com a numeração. Apenas tem de criar identificadores únicos para as suas referências. Para fazer isto, por vezes usam-se esquemas como `fig.etiqueta_da_figura`, ou `table.etiqueta_da_tabela`, embora o  $\text{\LaTeX}$  não imponha qualquer estilo de referência.

Em documentos grandes pode ser difícil ao autor lembrar-se dos nomes das etiquetas que usou. Durante a escrita (mas não na versão final!) do documento, a inclusão do comando `\usepackage{showlabels}` no preâmbulo do documento (antes de `\begin{document}`) faz com que sejam mostradas nas margens do documento os nomes das etiquetas que estão a ser usadas.

### Exercício 3.11

Edite o seu documento e coloque etiquetas em todos os objetos numerados que o mesmo possui. Note que não pode usar etiquetas iguais para objetos diferentes. Uma vez feita a atribuição de etiquetas, use o comando `\ref` e ou `\autoref` para obter e apresentar o seu número. Não se esqueça que o comando `\autoref` necessita da *package* `hyperref`. Inclua depois o comando `\usepackage{showlabels}` no preâmbulo do documento e veja o que mudou.

### 3.8.6 Disposição de elementos em matriz

A disposição de elementos alinhados segundo uma matriz é muitas vezes útil, nomeadamente para criar tabelas (mas não só). Para esse fim deve ser usado o ambiente `tabular`, em modo texto, ou `array`, em modo matemático. Vamos aqui abordar apenas o primeiro.

Para facilitar a explicação de como se usa o ambiente `tabular` observe-se o seguinte conteúdo  $\text{\LaTeX}$ :

```
\begin{tabular}{|l||c|r|} % Alinhamento: Esq, Centro, Dir
%
\hline
    & Temperatura & Humidade \\
Cidade & (\textordmasculine C) & (perc.) \\ \hline
Aveiro & {\large 10} & {\large 90} \\ \hline
Lisboa & {\tiny 13} & {\tiny 84} \\ \hline
Porto & \textbf{9} & \textbf{89} \\ \hline
%
\end{tabular}
```

que produz a seguinte matriz de elementos, ou tabela:

Cidade	Temperatura (°C)	Humidade (perc.)
Aveiro	10	90
Lisboa	13	84
Porto	9	89

O ambiente **tabular** produz uma caixa que, internamente, possui dados organizados de forma matricial. A matriz é especificada linha a linha, e a separação de linhas é indicada com o comando `\\`. O separador de elementos na mesma linha é o carácter `'&'`, um dos caracteres especiais referidos na Secção 3.5. As linhas e colunas podem ser opcionalmente separadas por linhas horizontais e verticais, respetivamente. A presença das linhas verticais é indicada através do parâmetro de formatação de colunas, nomeadamente através do símbolo `'|'`; a presença das linhas horizontais é indicada através de comando `\hline`. Por omissão os elementos de cada linha são alinhados pela sua base (veja a linha relativa a Aveiro).

O ambiente **tabular** serve para produzir tabelas ou matrizes de elementos mas não está de modo nenhum associado ao objeto móvel tabela. Este último não passa de uma área móvel com uma determinada legenda, referenciável, enquanto que o primeiro serve para dispor elementos espacialmente segundo uma matriz. Para ilustrar esta separação, repare que na pág. 3.10 usámos um ambiente **tabular** para apresentar as dimensões de letra, e não usámos um objeto flutuante **table**.

### 3.8.7 Expressões e ambientes matemáticos

As expressões matemáticas são interpretadas num contexto diferente do do texto normal, tendo regras diferentes de gestão do resultado produzido. As expressões matemáticas podem ser colocadas em ambientes de linha delimitados pelos caracteres `'$'`, ou em ambientes matemáticos mais alargados, que podem ocupar um espaço vertical variável.

Uma forma de criar um ambiente matemático é com o comando **\$\$**, que inicia e termina um ambiente matemático numa nova linha:

---

a expressão `$$y = ax^2 + bx + c$$` é a forma geral da equação de 2º grau

---



a expressão

$$y = ax^2 + bx + c$$

é a forma geral da equação de 2º grau

Note-se a diferença quando se usam apenas os caracteres **\$** isolados:

---

a expressão `$y = ax^2 + bx + c$` é a forma geral da equação de 2º grau

---



a expressão  $y = ax^2 + bx + c$  é a forma geral da equação de 2º grau

Outra forma de criar um ambiente matemático é recorrendo aos comandos `\[` e `\]`:

---

a expressão  
`\[`  
`y = ax^2 + bx + c`  
`\]`  
é a forma geral da equação de 2º grau

---



a expressão  
$$y = ax^2 + bx + c$$
  
é a forma geral da equação de 2º grau

Uma forma alternativa de criar um ambiente matemático com numeração é usando o ambiente `equation`:

---

a expressão `\ref{2ordem.eq}`  
`\begin{equation}`  
`y = ax^2 + bx + c \label{2ordem.eq}`  
`\end{equation}`  
é a forma geral da equação de 2º grau

---



a expressão 1  
$$y = ax^2 + bx + c \tag{1}$$
  
é a forma geral da equação de 2º grau

Uma forma alternativa de criar um ambiente matemático multilinha com numeração é usando o ambiente `eqnarray`:

---

Perímetro e área do círculo:

```
\begin{eqnarray}
P = 2\pi r \\
A = \pi r^2
\end{eqnarray}
```

---



Perímetro e área do círculo:

$$P = 2\pi r \quad (2)$$

$$A = \pi r^2 \quad (3)$$

Dois dos caracteres especiais do  $\text{\LaTeX}$  já antes referidos mas ainda não explicados são o  $\wedge$  e o  $\_$ . Como se viu acima nas expressões matemáticas, o  $\wedge$  serve para indicar um expoente, ou um elemento superior de um operador matemático, como um somatório ou um integral. Ao invés, o  $\_$  serve para indicar um índice, ou um elemento inferior de um operador matemático.

Vejam-se os dois exemplos abaixo para clarificar o seu uso, bem como as duas maneiras como podem ser produzidas as mesmas expressões matemáticas consoante tal acontece num ambiente de linha (primeiro caso) ou num ambiente matemático (segundo caso):

---

```
$y = \sum_{i=0}^{i=n}{x_{i}^{2n}}$
$$y = \sum_{i=0}^{i=n}{x_{i}^{2n}}$$
```

---



$$y = \sum_{i=0}^{i=n} x_i^{2n}$$

$$y = \sum_{i=0}^{i=n} x_i^{2n}$$

### 3.9 Inclusão de figuras

É normal os documentos incluírem imagens, porém o  $\text{\LaTeX}$  não possui nenhum mecanismo nativo de inclusão de imagens. Tal tem de ser feito através de *packages*.

Um dos *packages* mais usados para este efeito é o **graphicx**. Este *package* permite incluir imagens de vários tipos (bitmap, encapsulated postscript, etc.) e redimensioná-los. A inclusão de imagens pode ou não ser feita no contexto de um objeto flutuante (figura ou tabela).

O exemplo seguinte inclui uma imagem no texto a partir de um ficheiro **ua.pdf** sem usar um objeto móvel

---

Este  
`\includegraphics[height=24pt]{latex/ua}`  
é o novo logotipo da UA

---

e o resultado é: Este  é o novo logotipo da UA

Também é possível utilizar a mesma imagem dentro de um objeto flutuante. Este é possivelmente o caso mais utilizado:

---

```
\begin{figure}[h]
\center % Centra as imagens
\includegraphics[height=24pt]{latex/ua}
\caption{Novo logotipo da Universidade de Aveiro}
\label{fig:ualogo.1}
\end{figure}
```

---

e o resultado é a Figura 3.2.



Figura 3.2: Novo logotipo da Universidade de Aveiro

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 localizado no diretório `latex`). 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.

Este comando possui muito mais opções, sendo estas as consideradas mais úteis. Para mais pormenores deve ser consultada a documentação do pacote<sup>5</sup>. O exemplo seguinte, que resulta na Figura 3.3, mostra como utilizar a mesma imagem, processada de várias formas, com as opções indicadas.

---

<sup>5</sup>[ftp://ftp.di.uminho.pt/pub/ctan/macros/latex/required/graphics/grfguide.pdf](http://ftp.di.uminho.pt/pub/ctan/macros/latex/required/graphics/grfguide.pdf)

---

```

\begin{figure}[h]
\center % Centra as imagens
a) \includegraphics{latex/ua}
b) \includegraphics[height=3cm]{latex/ua}
c) \includegraphics[width=20mm]{latex/ua}
d) \includegraphics[scale=.5,angle=90]{latex/ua}
e) \includegraphics[height=5mm,width=3cm]{latex/ua}
\caption{Logotipo da Universidade de Aveiro: a) na dimensão real,
b) com 3cm de altura, c) com 20mm de largura, d) com altura e largura
reduzidas a  $\frac{1}{2}$  e simultaneamente rodado  $90^\circ$  e e) com uma modificação
anamórfica da altura e da largura.}
\label{fig:ualogo.2}
\end{figure}

```

---

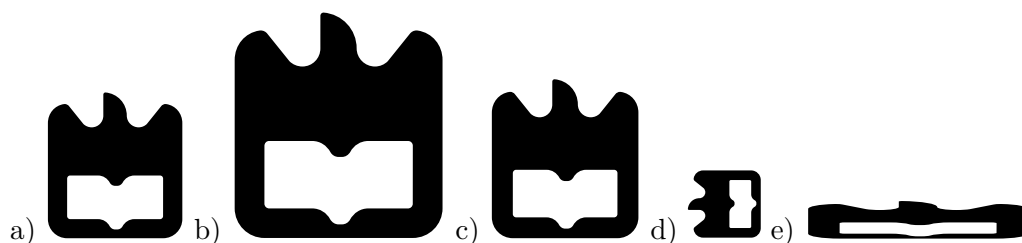


Figura 3.3: Logotipo da Universidade de Aveiro: a) na dimensão real, b) com 3cm de altura, c) com 20mm de largura, d) com altura e largura reduzidas a  $\frac{1}{2}$  e simultaneamente rodado  $90^\circ$  e e) com uma modificação anamórfica da altura e da largura.

### Exercício 3.12

Obtenha a partir da Internet várias imagens e guarde-as no formato PNG (Portable Network Graphics) ou PDF. Inclua estas figuras no seu documento, em vários locais (no texto, em objetos móveis) e redimensione-as. Observe o resultado no PDF resultante do documento.

## 3.10 Índices de conteúdos, de figuras e de tabelas

Por vezes é útil que os documentos incluam nalguma parte um índice de conteúdos, um índice de figuras ou um índice de tabelas. O comando `\tableofcontents` provoca a inclusão, no local onde está colocado, de um índice de todas as partes, capítulos, secções, etc., usadas no documento. Os comandos `\listoffigures` e `\listoftables` incluem um índice das figuras e tabelas, respetivamente, no local onde os comandos são usados.

O uso destes comandos, à semelhança do que acontece com o comando de referência `\ref`, obrigam normalmente a duas compilações de um documento para que os conteúdos apresentados pelas tabelas estejam coerentes com os conteúdos presentes no texto (títulos/legendas e respetivos números, páginas).



## 3.11 Referências bibliográficas

Na produção de documentos técnicos e científicos é fundamental que existam referências bibliográficas indicando trabalhos relacionados ou fontes onde se pode procurar mais informação sobre um assunto. Estas referências são agrupadas num capítulo/secção que normalmente se coloca no final do documento.

Há muitas maneiras de criar a bibliografia de um documento. Neste guião vamos explicar como a mesma se produz a partir de um ficheiro de referências guardadas num formato genérico.

### 3.11.1 Ficheiro de bibliografia, BibTeX

Um ficheiro de bibliografia é um ficheiro com a extensão **.bib** que possui registos de documentos num formato independente do documento que os cita. Um registo é criado por vários comandos com nomes distintos, mas que possuem a mesma estrutura base. A Tabela 3.2 apresenta alguns exemplos.

<b>@book</b>	livro
<b>@article</b>	artigo em revista
<b>@inproceedings</b>	artigo em ata de conferência ( <i>proceedings</i> )
<b>@masterthesis</b>	dissertação de mestrado
<b>@phdthesis</b>	dissertação de doutoramento
<b>@techreport</b>	relatório técnico
<b>@manual</b>	manual
<b>@misc</b>	algo de diferente

Tabela 3.2: Comandos para criação de registos bibliográficos.

Cada registo é especificado através do conteúdo que se coloca entre as chavetas, normalmente dividido por linhas para facilitar a sua leitura e compreensão por quem o está a editar.

Na Figura 3.4 é apresentado o conteúdo de um ficheiro com registos bibliográficos (neste caso, apenas 2). Cada registo descreve uma publicação, neste caso apenas livros, e indica vários dos seus atributos: autor, título, número de páginas, editor, ano de publicação e os seus dois International Standard Book Numbers (ISBNs). O texto indicado imediatamente após a primeira chaveta de cada registo, na primeira linha, é a sua etiqueta; esta etiqueta servirá para referenciar o registo sempre que for citado.

A ordem dos atributos é irrelevante (trocámos a ordem de autor e título nos dois registos propositadamente para demonstrar este facto). A indentação é igualmente irrelevante, apenas facilita a leitura pelo editor. Finalmente, o nome dos atributos pode usar minúsculas ou maiúsculas, como acontece no exemplo apresentado.

Para cada tipo de documento, livro, artigo, etc., existe um conjunto de atributos obrigatórios; existem também atributos opcionais. A lista de atributos é extensa e pode ser consultada em múltiplos documentos; recomenda-se uma consulta rápida na página do BibTeX na Wikipedia<sup>6</sup>.

<sup>6</sup><http://en.wikipedia.org/wiki/BibTeX>

---

```

@book{Kopka.03,
  title = {Guide to LaTeX},
  author = {Helmut Kopka and Patrick W. Daly},
  pages = {624},
  publisher = {Addison-Wesley Professional},
  edition = {4th},
  year = {2003},
  note = {ISBN-10: 0321173856, ISBN-13: 978-0321173850}
}

@book{Lamport.94,
  AUTHOR = {Leslie Lamport},
  TITLE = {LaTeX: A Document Preparation System},
  PAGES = {288},
  PUBLISHER = {Addison-Wesley Professional},
  EDITION = {2nd},
  YEAR = {1994},
  NOTE = {ISBN-10: 0201529831, ISBN-13: 978-0201529838}
}

```

---

Figura 3.4: Exemplo de ficheiro de registos BibTeX contendo dois registos.

Note que a separação de nomes de autores no atributo **AUTHOR** é feita com a palavra **and** e não com vírgulas, como é normal. Tal é feito porque o nome de um autor pode ser colocado com o apelido primeiro, usando-se a vírgula para os separar:

```

AUTHOR = {Lamport, Leslie}

AUTHOR = {Kopka, Helmut and Daly, Patrick W.}

```

### 3.11.2 Uso de citações bibliográficas

Quando se pretende citar um documento noutro segue-se o seguinte procedimento:

1. Coloca-se num ficheiro de bibliografia (**myBib.bib**, por exemplo) os registos dos documentos a citar (pressupomos os registos da Figura 3.4).
2. No documento que faz a citação usa-se o comando `\cite{Lamport.94}` para criar uma citação do livro de L. Lamport nesse local. Esta ação pode ser feita inúmeras vezes ao longo do documento, sempre com o mesmo resultado prático.
3. No final do documento colocam-se os comandos para incluir (automaticamente) a bibliografia e indica-se quais são os ficheiros que deverão ser considerados como fontes de registos bibliográficos. No nosso caso colocaríamos os seguintes comandos:

---

```

\bibliographystyle{plain}
\bibliography{myBib}

```

---

O primeiro comando, `\bibliographystyle`, indica o estilo que deverá ser usado para formatar as citações e as referências bibliográficas (neste caso **plain**, ver secção 3.11.3). O segundo comando, `\bibliography`, cria um capítulo ou secção com a bibliografia, ao mesmo tempo que indica uma lista de nomes de ficheiros, separados por vírgulas, onde se podem obter os registos das citações realizadas ao longo do documento e que devem ser incluídos nessa bibliografia. Note que os registos bibliográficos presentes nos ficheiros **.bib** que não são citados no documento são omitidos.

Feito isto, a bibliografia no documento é produzida do seguinte modo:

1. Compila-se o ficheiro **.tex** (**doc.tex**, por exemplo) de forma a gerar o documento no formato final (PDF, por exemplo):

```
$ pdflatex doc
```

2. Compila-se o ficheiro **.aux** (**doc.aux**, neste exemplo) com o comando **bibtex** de forma a gerar um ficheiro de bibliografia (neste caso, **doc.bbl**):

```
$ bibtex doc
```

Neste processo é usado transparentemente o ficheiro **myBib.bib** para fornecer os registos de documentos citados em **doc.tex**. O formato dos conteúdos colocados no ficheiro **.bbl** resultante depende do estilo usado no comando `\bibliographystyle`.

3. Compila-se novamente o ficheiro **.tex** de forma a gerar o documento no formato final mas já com as referências bibliográficas corretamente formatadas e incluídas.

```
$ pdflatex doc
```

### 3.11.3 Estilos de bibliografia

Há inúmeros estilos de formatação da bibliografia e das referências colocadas ao longo do texto, a maior parte das quais fornecidas através de pacotes. O BibTeX possui apenas 4 estilos base:

**plain**: as referências são ordenadas alfabeticamente pelo último nome do primeiro autor e são identificadas por um número sequencial começando em 1.

**unsrt**: as referências são ordenadas pela ordem em que são pela primeira vez referenciadas no texto e são identificadas por um número sequencial começando em 1.

**abbrv**: similar ao **plain** mas as referências possuem os nomes e alguns outros campos abreviados.

**alpha**: similar ao **plain** mas as referências são identificadas por um acrónimo formado pelas primeiras letras do último nome de alguns dos primeiros autores e pelos dois algarismos menos significativos do ano de publicação.

Se considerarmos o seguinte trecho de um documento L<sup>A</sup>T<sub>E</sub>X:

---

O `{\LaTeX}`~`\cite{Lamport.94}` é um sistema criado por Leslie Lamport e há inúmeros manuais detalhados sobre o mesmo (v.g.~`\cite{Kopka.03}`).

---

este texto e a bibliografia produzida tendo em conta estes 4 estilos apresentados e os registos da Figura 3.4 estão patentes na Figura 3.5.

<p><b>plain</b></p> <p>O <math>\LaTeX</math> [1] é um sistema criado por Leslie Lamport e há inúmeros manuais detalhados sobre o mesmo (v.g. [2]).</p> <p style="text-align: center;"><b>Bibliografia</b></p> <p>[1] Helmut Kopka and Patrick W. Daly. <i>Guide to <math>\LaTeX</math></i>. Addison-Wesley Professional, 4th edition, 2003. ISBN-10: 0321173856, ISBN-13: 978-0321173850.</p> <p>[2] Leslie Lamport. <i>LaTeX: A Document Preparation System</i>. Addison-Wesley Professional, 2nd edition, 1994. ISBN-10: 0201529831, ISBN-13: 978-0201529838.</p>	<p><b>unsrt</b></p> <p>O <math>\LaTeX</math> [1] é um sistema criado por Leslie Lamport e há inúmeros manuais detalhados sobre o mesmo (v.g. [2]).</p> <p style="text-align: center;"><b>Bibliografia</b></p> <p>[1] Leslie Lamport. <i>LaTeX: A Document Preparation System</i>. Addison-Wesley Professional, 2nd edition, 1994. ISBN-10: 0201529831, ISBN-13: 978-0201529838.</p> <p>[2] Helmut Kopka and Patrick W. Daly. <i>Guide to <math>\LaTeX</math></i>. Addison-Wesley Professional, 4th edition, 2003. ISBN-10: 0321173856, ISBN-13: 978-0321173850.</p>
<p><b>abbrv</b></p> <p>O <math>\LaTeX</math> [1] é um sistema criado por Leslie Lamport e há inúmeros manuais detalhados sobre o mesmo (v.g. [2]).</p> <p style="text-align: center;"><b>Bibliografia</b></p> <p>[1] H. Kopka and P. W. Daly. <i>Guide to <math>\LaTeX</math></i>. Addison-Wesley Professional, 4th edition, 2003. ISBN-10: 0321173856, ISBN-13: 978-0321173850.</p> <p>[2] L. Lamport. <i>LaTeX: A Document Preparation System</i>. Addison-Wesley Professional, 2nd edition, 1994. ISBN-10: 0201529831, ISBN-13: 978-0201529838.</p>	<p><b>alpha</b></p> <p>O <math>\LaTeX</math> [Lam94] é um sistema criado por Leslie Lamport e há inúmeros manuais detalhados sobre o mesmo (v.g. [KD03]).</p> <p style="text-align: center;"><b>Bibliografia</b></p> <p>[KD03] Helmut Kopka and Patrick W. Daly. <i>Guide to <math>\LaTeX</math></i>. Addison-Wesley Professional, 4th edition, 2003. ISBN-10: 0321173856, ISBN-13: 978-0321173850.</p> <p>[Lam94] Leslie Lamport. <i>LaTeX: A Document Preparation System</i>. Addison-Wesley Professional, 2nd edition, 1994. ISBN-10: 0201529831, ISBN-13: 978-0201529838.</p>

Figura 3.5: Exemplo do aspeto final da bibliografia consoante o estilo usado. A etiqueta de cada registo é exatamente igual ao que aparecerá ao longo do texto, no local onde for feita a respetiva referência.

### Exercício 3.13

Edite um ficheiro com registos bibliográficos, coloque lá um de cada tipo com um número mínimo de atributos (autor, título e ano chegam, pode inventar). Inclua no seu documento  $\text{\LaTeX}$  os comandos de inclusão de bibliografia e escolha um estilo. Inclua referência no seu texto ao registos bibliográficos que criou e gere as respetivas referências com o comando BibTeX. Observe as mensagens de erro (atributos em falta) reportados pelo BibTeX e atribua-os. Proceda até conseguir ter um documento que cite todas registos bibliográficos que criou. Altere o estilo de formatação das citações e observe o resultado.

## 3.12 Visão global da geração de documentos $\text{\LaTeX}$

Para terminar este documento vamos dar uma visão global do modo como o  $\text{\LaTeX}$  deve ser usado para produzir um documento complexo. Esta explicação será ilustrada pela Secção 3.12. Os ficheiros fonte estão a azul, o ficheiro alvo está a preto, os ficheiros verdes são ficheiros temporários, criados automaticamente, que são fundamentais para o processo de geração do documento alvo mas que podem em qualquer altura ser apagados. As setas a vermelho apontam para resultados de comandos realizados usando como parâmetro o ficheiro no início da seta. As setas tracejadas a azul representam inclusões de ficheiros por referência a partir de outros ficheiros, sendo indicado o comando responsável por essa inclusão.

O texto fonte de um documento pode estar repartido por vários documentos fonte, todos com a extensão `.tex`. Na Secção 3.12 são usados 4, um principal (`doc.tex`) e 3 outros incluídos por este usando o comando `\input`.

Para além deste ficheiros, o texto fonte de um documento existe também ao nível dos ficheiros de registos bibliográficos usados para retirar referências usadas no documento. Na Secção 3.12 são usados 3, os quais são associados ao documento através do comando `\bibliography`.

Resumindo, o documento possui 7 ficheiros fonte, 4 com conteúdos indiscriminados e 3 com registos BibTeX.

A compilação do documento faz-se com o comando `pdflatex` passando como parâmetro o nome base do ficheiro principal do documento (`doc.tex` ou `doc`, a extensão pode ser omitida). Daqui resulta um documento PDF (`doc.pdf`), que é o objetivo pretendido, e mais uma série de ficheiros que são um subproduto da compilação.

Alguns destes ficheiros são meramente informativos, como o `doc.log`, que possui um registo das atividades executadas pelo comando, ficheiros usados, erros identificados, etc.



### 3.13 Para aprofundar

#### Exercício 3.14

O  $\text{\LaTeX}$  possui muitas outras funcionalidades não exploradas aqui. Um bom exemplo é a capacidade de desenhar gráficos sem necessidade de recorrer à importação de ficheiros.

Explore a *package tikz* e verifique como pode construir imagens directamente em  $\text{\LaTeX}$ . Consulte o endereço <http://www.texample.net/tikz/examples>.

### Glossário

<b>ASCII</b>	American Standard Code for Information Interchange
<b>ISBN</b>	International Standard Book Number
<b>PDF</b>	Portable Document Format
<b>PDF</b>	Portable Document Format
<b>WYSIWYM</b>	What You See Is What You Mean
<b>WYSIWYG</b>	What You See Is What You Get

### Referências

- [1] Wikimedia, *Latex - Wikibooks, Open books for an open world*, <http://en.wikibooks.org/wiki/LaTeX>, [Online; acedido em 12 de Outubro de 2016], 2013.
- [2] Stack Exchange, *TeX - LaTeX Stack Exchange*, <http://tex.stackexchange.com>, [Online; acedido em 12 de Outubro de 2016], 2013.
- [3] V. Cerf, *ASCII format for network interchange*, RFC 20, Internet Engineering Task Force, out. de 1969.