

LATEX

UM SISTEMA DE PREPARAÇÃO DE TEXTOS

Editora Virtual Imhotep, Ltda.

Dedicado a todos os amantes da tipografia.

Sumário

<i>Introdução</i>	11
<i>Criando um texto básico</i>	15
<i>A estrutura do documento</i>	21
<i>Tabelas</i>	27
<i>Formatação</i>	29
<i>Figuras</i>	39
<i>Referências</i>	45
<i>Cores</i>	47
<i>Fontes</i>	49
<i>Idiomas</i>	51
<i>Outros recursos</i>	55

Lista de Figuras

- | | | |
|---|-------------------------------------------------------|----|
| 1 | Um texto simples. | 16 |
| 2 | Gustave Courbet, <i>Femme à la Vague</i> (1868). | 40 |
| 3 | Jean-Baptiste Corot. <i>Ville d'Avray</i> (ca. 1867). | 42 |
| 4 | <i>L'apparition</i> , de Gustave Moreau (1875). | 43 |

Lista de Tabelas

1	Principais opções das classes padrão do L ^A T _E X	18
2	Comandos de seção	24
3	Exemplo de tabela	27
4	Comandos utilizados para tabelas	28
5	Aspas	32
6	Estilos de fonte	33
7	Tamanho do texto	34
8	Símbolos comuns	35
9	Signos astrológicos e do zodíaco	35
10	Alinhamento de parágrafos	35
11	As 68 cores da opção dvipsnames	47

Introdução

O que é TeX?

TeX(X ou chi pronunciado como o j espanhol ou k) é uma linguagem de mark-up e de programação de baixo nível criada por Donald Knuth para diagramar documentos com uma estética agradável e estruturalmente consistentes. O nome TeX é a primeira sílaba das palavras gregas τεχνολογία (“tecnologia”), e τέχνη (*techni*), “arte”.

Quando Knuth começou a escrever o programa de diagramação (*typesetting engine*) em 1977 para explorar o potencial dos equipamentos de impressão digital que estavam começando a se infiltrar na indústria da editoração gráfica na época, esperando especialmente reverter a tendência da deterioração da qualidade tipográfica afetava seus próprios livros e artigos.

O TeX é uma linguagem de programação para diagramação de textos (*typesetting*). O controle fino que o TeX oferece o torna uma ferramenta muito poderosa, mas também muito difícil, consumindo muito tempo. Mas TeX é extremamente estável, pode rodar em muitos tipos de computador, e é praticamente livre de bugs.

Baseado em linguagem TeX, na década de 1980 Leslie Lamport criou o conjunto de macros chamado L^AT_EX, que poupa tempo, automatiza certas tarefas e reduz erros, especialmente para documentos que contenham fórmulas matemáticas.

O L^AT_EX é um software modular, composto de programas binários e conjuntos de macros chamados pacotes ou estilos, que podem ser utilizados, acrescentados e modificados. Alguns destes são empacotados com a maioria das distribuições do TeX/L^AT_EX; outros podem ser encontrados no *Comprehensive TeX Archive Network* (CTAN).

Como o L^AT_EX inclui um grupo de comandos TeX, o processamento de um documento L^AT_EX é essencialmente programação. Você cria um arquivo de texto em marcação L^AT_EX. O programa “lê” este texto marcado para produzir o documento final. Essa abordagem tem algumas desvantagens em comparação com programas visuais como o OpenOffice/LibreOffice e o Microsoft Word®.

Por que usar o L^AT_EX?

Devemos ser muito claros: você usa o processador de texto que quiser. Se quiser escrever seus textos e entregá-los com o Word, ou OpenOffice, ou Pages da Apple, ou o InDesign, da Adobe, ou GoogleDocs, para o conteúdo do seu texto isso vai fazer pouca diferença. Mas se você quer uma aparência profissional, usa muitas figuras ou fórmulas matemáticas, está escrevendo uma tese um ou

texto técnico ou acadêmico, o resultado final, com o mesmo nível de esforço, vai ser muito diferente. No começo o L^AT_EX parece muito difícil, mas facilita e automatiza muitas tarefas complexas, e a maioria dos comandos é bem intuitiva. Na verdade, ele não passa de um conjunto de macros para geração de texto. E, uma vez que você personaliza um modelo, você simplesmente “joga” qualquer texto nele que ele sai com a aparência desejada. Com os outros programas, acontece o oposto: você tem mais liberdade, deve saber muito bem o que está fazendo. O L^AT_EX, na verdade, faz o trabalho do diagramador ou designer gráfico para você.

Contudo, você não precisa, necessariamente, usar esta ferramenta. Devo deixar claro que, no Brasil, praticamente nenhuma publicação aceita esse formato. Mas temos que salientar que o resultado é profissional, para impressão: compare um livro bem produzido e um artigo acadêmico pobemente formatado no Word: o L^AT_EX é a maneira mais fácil e barata de produzir textos que mais se aproximam de revistas e livros compostos com um profissional (muito bem) pago para cuidar da aparência do seu material.

Porém, o L^AT_EX tem alguns inconvenientes:

- você geralmente não vê a versão final do documento quando está editando;
- você geralmente precisa de conhecer os comandos de marcação;
- pode ser difícil dar um visual muito personalizado aos documentos.

Por outro lado, há certas vantagens na abordagem L^AT_EX:

- o programa é de graça, de código aberto, com manuais que explicam praticamente tudo;
- os documentos originais podem ser lidos em qualquer editor de texto;
- você pode se concentrar exclusivamente na estrutura e no conteúdo do documento, sem ficar perdido em questões superficiais de layout;
- você não precisa de ajustar manualmente as fontes, o tamanho do texto, espaçamento e divisão de linhas para legibilidade: o L^AT_EX cuida disso automaticamente;
- o layout, as fontes, as tabelas etc. são consistentes no documento todo;
- fórmulas matemáticas podem ser escritas com facilidade;
- índices, notas de rodapé, citações e referências são geradas com facilidade.
- você é forçado a estruturar seus documentos corretamente.

Como o L^AT_EX funciona

1. Você digita um texto
2. faz as marcações necessárias (indicando título, capítulo, seções, ênfases etc.)
3. roda o programa para gerar o produto final (um texto diagramado em .pdf ou outros formatos menos usuais).

Para isso, você precisa de, no mínimo, uma distribuição (uma versão do programa) e um editor de textos específico, (para facilitar a digitação e compilação do texto), além de um leitor de pdfs (Acrobat Reader ou similares).

Instalando o programa (uma “distribuição”)

Há dois tipos básicos de distribuição:

TeXLive (<http://www.tug.org/texlive/>) para Linux, Mac (<http://www.tug.org/mactex>) e Windows. No Linux, pode instalar a partir da linha de comando (por exemplo, no Ubuntu: `sudo apt-get install texlive-full`), ou através do gerenciador de pacotes.

MiKTeX (<http://www.miktex.org>) para Windows.

Instalando um editor de textos

Há muitos por aí. Na verdade, você pode editar seus textos no Notepad ou até mesmo no Word ou no OpenOffice/LibreOffice. Eu vou citar apenas alguns.

Texworks (<http://www.tug.org/texworks/>). Vem com MiKTeX e com Texlive para Windows e Mac; no Linux tem que instalar separadamente.

TexShop (<http://pages.uoregon.edu/koch/texshop/>), para Mac, vem com MacTeX.

Kile (<http://kile.sourceforge.net>) É um editor para o ambiente gráfico KDE do Linux.

TeXmaker (<http://www.xm1math.net/texmaker>) É um editor útil multiplataforma com seu próprio leitor de pdfs.

TeXStudio (<http://texstudio.sourceforge.net>) Muito semelhante ao TeXmaker, aceita também correção ortográfica (o mesmo dicionário do OpenOffice). Tem também uma grande quantidade de símbolos matemáticos e funções de autocompletar que agilizam a criação do texto.

Criando um texto básico

Um arquivo .tex tem basicamente duas partes: o *préâmbulo*, que vem antes do texto e define aspectos como tamanho da letra, aparência, layout das margens e tamanho da página, e o *texto* propriamente dito.

A melhor forma de aprender o L^AT_EX é testando e observando o código (o arquivo original antes de ser compilado/diagramado). Aqui veremos como criar um artigo simples. Programas como o TexStudio, que foi utilizado para compor este texto, contêm préâmbulos pré-definidos, que, com algumas alterações, podem ser usados para compor um texto em português com o mínimo de esforços. Mas é claro que você pode adicionar seu toque pessoal, à medida em que você for aprendendo. Aqui veremos um exemplo básico, então não vamos nos preocupar com fontes (tipos de letras) ainda. No entanto, valem algumas considerações iniciais.

Vejamos agora como fica um texto básico, tirado da Wikipedia:

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}

\title{O Sistema \LaTeX}
```

```
\begin{document}
```

```
\maketitle
```

A ideia central do \LaTeX\ é distanciar o autor
o máximo possível da apresentação visual da informação.

Ao invés de trabalhar com ideias visuais, o usuário é
encorajado a trabalhar com conceitos mais lógicos --- e,
consequentemente, independente da apresentação --- como capítulos,
seções, ênfase e tabelas, sem contudo impedir o usuário da
liberdade de indicar, expressamente, declarações de formatação.

A versão mais recente é a \LaTeXe.

```
% Isto é um comentário que não será processado. Ele serve apenas
% para fazer anotações não incluídas no resultado final. Atenção
% ao símbolo do comentário: porcentagem (%).
```

A seguir, a fórmula das combinações como um exemplo simplório

da capacidade matemática do \LaTeX:

```
\begin{eqnarray}
_{\text{k}^{\text{n}}}\text{ } \&=& \frac{\text{n}!}{\text{k!}(n-\text{k})!} \\
\end{eqnarray}
\end{document}
```

Isso vai ter a aparência da figura 1.

O Sistema \LaTeX

Figura 1: Um texto simples.

9 de novembro de 2011

A ideia central do \LaTeX é distanciar o autor o máximo possível da apresentação visual da informação.

Ao invés de trabalhar com ideias visuais, o usuário é encorajado a trabalhar com conceitos mais lógicos — e, consequentemente, independente da apresentação — como capítulos, seções, ênfase e tabelas, sem contudo impedir o usuário da liberdade de indicar, expressamente, declarações de formatação.

A versão mais recente é a \BIBTeX 2ε.

A seguir, a fórmula das combinações como um exemplo simplório da capacidade matemática do \LaTeX:

$$C_k^n = \frac{n!}{k!(n-k)!} \quad (1)$$

No caso, tudo o que eu precisei de fazer foi rodar o comando `pdflatex latexbasico.tex` na pasta onde deixei o arquivo `latexbasico.tex` ou então apertar o botão `pdflatex` no menu do editor para gerar um arquivo `pdf` de mesmo nome. Depois é só abrir o `pdf` e ver como ficou.

Comandos

Os comandos em \LaTeX são sensíveis a maiúsculas e minúsculas, e têm um dos dois formatos:

- Começam com um \ e têm um nome composto só de letras. Nomes de comando são terminados por um espaço, por um número ou qualquer outro caractere que não é letra.
- Consistem de um \ e exatamente um caractere não alfabético.

Ambientes \LaTeX

Os *ambientes* têm um papel semelhante a comandos, mas geralmente afetam só uma parte específica do documento.

```
\begin{NOME DO AMBIENTE}
Texto a ser influenciado
\end{NOME DO AMBIENTE}
```

Entre \begin{ e \end você pode colocar outros comandos e outros ambientes. Todo no \LaTeX pode ser expresso em termos de comandos e ambientes.

Caracteres especiais

Como a rigor não existe interface gráfica para um arquivo .tex (que é de texto simples), alguns caracteres são usados para os comandos:

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

O que fazer, então, quando quisermos que eles apareçam no texto? Geralmente colocamos uma barra à esquerda, ou outros comandos quando o caso é a própria barra à esquerda (que se usarmos duas, gera um novo parágrafo) ou o acento circunflexo:

\#	#
\\$	\$
\%	%
\textasciicircum{}	^
\&	&
_	_
\{ \}	{ }
\~{}	~
\textbackslash{}{}	\

Comentários

Podemos escrever um texto, “comentando” o original, mas esse comentário não aparecerá no texto compilado. Para isso, usamos %.

Espaços, acentos e pontuação

Espaços maiores que um espaço são desconsiderados (mas há comandos para criar espaços maiores). Para começar outro parágrafo, tecle duas vezes ENTER ou \\. (Se teclar mais que duas vezes o enter o resultado será o mesmo.)

Para inserir caracteres acentuados diretamente, como em português, basta inserir

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

ou

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

no preâmbulo.

Para inserir um travessão (—), use três traços: ---, e para separar números ou datas, use -- (significando: “de – até”).

Para aspas inglesas, use: ‘‘ para abrir e ’’ para fechar aspas.

Classes de documentos

A primeira informação que o L^AT_EX precisa de saber ao processar um arquivo é o tipo de documento que o autor quer criar. Isso é especificado usando o comando \documentclass{...}.

A classe é o tipo de documento que deve ser criado. A distribuição L^AT_EX fornece classes adicionais para outros documentos, incluindo cartas e slides. Os

parâmetros opcionais personalizam o comportamento da classe, e estão entre colchetes, devendo ser separados por vírgulas.

Exemplo: um arquivo de entrada (input) pode começar com:

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

Que significa: o documento é um artigo, com fonte básica de tamanho de 11 pontos (11pt), página frente e verso (twoside) e papel A4.

Algumas outras classes são: book (livro), report, abnt (formato ABNT), beamer (para apresentações de slides). Outras opções podem ser vistas na tabela 1.

10pt, 11pt, 12pt	Tamanho da fonte principal do documento. Geralmente o tamanho padrão das classes é 11pt
a4paper, letterpaper, a5paper, legalpaper, executivepaper...	Tamanho do papel. Geralmente vem com A4, se não for especificado.
fleqn	Fórmulas alinhadas à esquerda.
leqno	Número da fórmula à esquerda.
titlepage, notitlepage	Especifica se vai ter ou não página separada para o título. A classe article não tem, por padrão; report e book, têm. Se quiser mudar esse comportamento padrão, deve indicar
onecolumn, twocolumn	Uma ou duas colunas.
twoside, oneside	twoside=frente e verso, padrão em livros; oneside=só frente, padrão em article e report.
landscape	página na horizontal
openright, openany	Começa novos capítulos com páginas da direita (openright) ou qualquer página (openany). Padrão para livros: openright.
draft	Esboço. Indica problemas de hifenização e justificação com um pequeno quadrado na margem para ser localizado.

Tabela 1: Principais opções das classes padrão do L^AT_EX

Por exemplo, se quiser escrever um relatório com fonte 12 pontos em A4, em modo de esboço, você usaria:

```
\documentclass[12pt,a4paper,draft]{report}
```

Pacotes

Ao escrever um documento, você vai perceber que provavelmente há algumas áreas em que o básico não vai resolver seu problema. Se quiser colocar figuras, texto colorido, fontes, personalizar margens, você vai precisar de “melhorias”. Essas melhorias são chamadas de “pacotes”. Eles são ativados com o comando

```
\usepackage[opções]{pacote}
```

em que “opções” são palavras chave que ativam características especiais do pacote e “pacote” é o nome do pacote. A maioria das distribuições atuais vêm com muitos pacotes e classes, mas mesmo assim é possível instalá-los manualmente. O Miktex tem uma opção de instalar automaticamente pacotes que faltam (se estiverem disponíveis nos servidores), e o Texlive também tem uma função semelhante (ver os sites da seção na página 13). Uma maneira mais fácil é colocar estilos baixados da internet (com extensões .sty para pacotes e .cls para classes) na mesma pasta em que está o documento a ser processado, mas isso tem o inconveniente de que só documentos naquela mesma pasta poderão usar os estilos. Para instalar no sistema, quando estamos usando Texlive, podemos criar uma pasta chamada `texmf` no diretório do usuário `~/[NOME DO USUÁRIO]` no Linux ou `C:\Users\[NOME DO USUÁRIO]` no Windows. Então, vamos até essa pasta no prompt de comando e digitamos: `texhash texmf`. É assim que podemos instalar, com o Texlive, o estilo ABNT (ver a página do ABNTeX2 em: <http://code.google.com/p/abntex2/>.)

A estrutura do documento

O principal objetivo de um texto é transmitir ideias, informação ou conhecimento ao leitor. O leitor entenderá melhor o texto se essas ideias forem bem estruturadas, e verá e sentirá essa estrutura melhor se a forma tipográfica refletir a estrutura lógica e semântica do conteúdo.

O L^AT_EX é diferente de outros sistemas de diagramação, já que você só precisa de informar a estrutura lógica e semântica do texto. Então, o programa deriva a forma tipográfica de acordo com as “regras” dadas na classe do documento e nos vários estilos, permitindo ao usuário estruturar seu documento de acordo com várias construções hierárquicas, incluindo capítulos, seções, subseções e parágrafos.

O ambiente document

Depois da declaração da classe do documento, o texto de seu documento encontra-se entre dois comandos que identificam o começo e o fim do documento propriamente dito:

```
\documentclass[a4paper]{report}

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

Os três pontos indicam onde você coloca o texto. A razão para marcar o começo do texto permitir configurações extras (ver a linha branca no exemplo acima: vamos utilizá-la em breve).

Preâmbulo

O *preâmbulo* é tudo entre o começo do arquivo-fonte até `\begin{document}`. Normalmente contém comandos que afetam todo o documento.

```
% simple.tex - Um artigo simples para ilustrar a estrutura do documento.
```

```
\documentclass{article}
\usepackage{mathptmx}
\usepackage[brazil]{babel}
\usepackage[utf8]{inputenc}

\begin{document}
```

A primeira linha, iniciada com %, é um comentário (não vai aparecer no texto final). O comando `\documentclass` indica o tipo de documento que queremos produzir. O pacote `babel` serve para indicar as línguas usadas (no caso, `brazil` = português do Brasil). `\usepackage[utf8]{inputenc}` indica uso de caracteres acentuados diretamente e `\usepackage{mathptmx}` indica o uso da fonte Times ao invés da fonte padrão do L^AT_EX, que é a Computer Modern (trataremos as fontes na seção mais adiante).

Todos os “pacotes” que serão tratados daqui em diante, devem ser inseridos no preâmbulo (isto é, antes de `\begin{document}`). Assim, para usar o pacote `enumitem` (ver página 30), devemos inserir no preâmbulo: `\usepackage{enumitem}`.

O cabeçalho

No começo da maioria dos documentos haverá informação sobre o próprio documento: título, data, autor(es) etc.

Um exemplo simples:

```
\documentclass[12pt,a4paper]{report}

\begin{document}
\title{Como estruturar um documento em \LaTeX{}}
\author{Fulano de Tal}
\date{Dezembro de 2012}
\maketitle
\end{document}
```

Os comandos `\title`, `\author` e `\date` são autoexplicativos. Se não for indicada a data, aparecerá o dia em que o documento foi composto. Você sempre termina o cabeçalho com `\maketitle`, indicando que você terminou o cabeçalho e que ele pode ser diagramado de acordo com a classe (estilo) que você escolheu. Se você omitir `\maketitle`, essas informações não aparecerão (a não ser que você as escreva manualmente).

Aqui temos um exemplo mais complicado:

```
\title{Como estruturar um documento em \LaTeX{}}
\author{Fulano de Tal\\
    Escola de Computação\\
    Universidade Tal,\\
    São Paulo,\\
    Brasil,\\
    \texttt{fulanodetal@universidadetal.br}}
\date{\today}
\maketitle
```

Como pode ver, pode usar os argumentos `\title` e os outros. A barra dupla à esquerda (\\\) é o comando para forçar uma quebra de linha. O L^AT_EX normalmente decide por si só onde quebrar uma linha, e geralmente está certo, mas às vezes você precisa de terminar uma linha antes do fim, como aqui, e começar uma nova. No exemplo, na última linha, usamos uma fonte monoespaciada (de máquina de escrever) com o comando `\texttt{}`, para escrever o e-mail. O comando `\today` será substituído pela data do dia, mas você pode colocar a data

que quiser, sem ordem específica. Esses comandos do cabeçalho também não têm ordem específica (título, autor, data), eles serão processados de acordo com o estilo.

Se houver dois autores, pode-se separá-los com o comando `\and`:

```
\author{Fulano de Tal \and Sicrano de Tal}
```

Abstract/Resumo

Como a maioria dos artigos científicos têm um resumo (e um abstract em inglês, caso a língua original não seja esta), há comandos predefinidos para compor um abstract. O comando `abstract` está disponível nas classes `article` e `report`, mas não em `book`.

```
\begin{abstract}
Escreva seu resumo aqui
...
\end{abstract}
```

Para escrever o resumo (em português, por exemplo) e também o abstract (em inglês), você tem duas opções. Se estiver usando o pacote `babel`, através do comando

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

A língua ativada é sempre a última (no caso, `brazil=português do Brasil`). Você pode copiar esse código e salvar como um arquivo `.tex`, e fazer o teste você mesmo rodando o comando `pdflatex (nome do arquivo).tex`:

```
\documentclass{article}
\usepackage[english,brazil]{babel}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{utopia}

\title{Um texto qualquer}
\author{Fulano de Tal}
```

```
\begin{document}
```

```
\maketitle
```

```
\begin{abstract}
Um resumo em português.
\end{abstract}
```

```
\begin{otherlanguage}{english}
```

```
\begin{abstract}
This is an abstract in English.
\end{abstract}
```

```
\end{otherlanguage}

\end{document}
```

A outra opção é se estiver utilizando o pacote `polyglossia`:

```
\documentclass{article}
\usepackage{fontspec}
\setmainfont{Minion Pro}
\setsansfont{Myriad Pro}
\setmonofont{Courier}
\usepackage{polyglossia}
\setmainlanguage{brazil}
\setotherlanguages{french,english,german,norsk}
```

Abstract em inglês:

```
\begin{english}
\begin{abstract}
This is an abstract in English.
\end{abstract}
\end{english}
```

Comandos de seção

Os comandos para quebra de seção são bem intuitivos. Claro que alguns comandos são apropriados a diferentes classes de documentos.

Aqui temos um exemplo simples:

```
\section{Título da Seção}

\section{Outra seção}

\subsection{Uma subseção}
```

Note que não é necessário especificar números de seção, o L^AT_EX fará isso para você. Você também não precisa de marcar o começo e o fim de uma seção. O L^AT_EX dá 7 níveis de seção:

Comando	Nível	Comentário
<code>\part{PARTE}</code>	-1	não em cartas
<code>\chapter{CAPÍTULO}</code>	0	somente livros e relatórios
<code>\section{SEÇÃO}</code>	1	
<code>\subsection{SUBSEÇÃO}</code>	2	
<code>\subsubsection{SUBSUBSEÇÃO}</code>	3	
<code>\paragraph{PARÁGRAFO}</code>	4	
<code>\ subparagraph{SUBPARÁGRAFO}</code>	5	

Tabela 2: Comandos de seção

Todos os títulos são acrescentados automaticamente ao sumário (table of contents), se você decidir inserir um. Mas se você fizer mudanças manuais de

estilo no seu cabeçalho, por exemplo, um título muito longo, ou quebras de linha, ou uso incomum de fontes, isso também aparecerá no sumário — o que você provavelmente não quer. O L^AT_EX lhe permite criar uma versão opcional extra que só será utilizada no sumário e no topo de página (se houver). Essa alternativa é colocada entre colchetes:

```
\section[título curto]{título longo}
```

Numeração de seções

O L^AT_EX numera automaticamente as seções. Partes têm numerais romanos (Parte I, Parte II), capítulos e seções, numeração decimal, e apêndices, alfabetica (A, B, C etc.). Você pode mudar a numeração, de acordo com a tabela 2. Assim, se você quiser numerar até as seções, mas não as subseções:

```
\setcounter{secnumdepth}{1}
```

Se você quiser uma seção/capítulo etc. não numerada que não vai aparecer no sumário, coloque um asterisco antes da chave:

```
\chapter*{Introdução}
```

Se você quiser que essa seção não numerada apareça no conteúdo/sumário, faça o seguinte:

```
\chapter*{Introdução}
\addcontentsline{toc}{chapter}{Introduction}
```

Apêndices

O comando `\appendix` indica que os capítulos ou seções devem ser numerados como apêndices:

```
\appendix
\chapter{Primeiro apêndice}
```

Sumário

Todos os cabeçalhos numerados automaticamente vão para o sumário. Você não precisa de ter um, mas se quiser, é só adicionar o comando `\tableofcontents` onde quiser que apareça o sumário (geralmente depois do abstract).

Os comandos `\listoffigures` e `\listoftables` geram, respectivamente, uma lista das figuras e uma lista das tabelas.

Bibliografia

A bibliografia, é claro, é uma das partes mais importantes (e mais chatas, e mais pedantes) de um texto científico. Existem centenas de estilos de citação e bibliografia. O L^AT_EX pode compor/gerar a bibliografia para você, com alguns comandos específicos (`bibtex`) e depois que você tiver um arquivo separado (`.bib`) para gerar essa bibliografia. Gerar um arquivo desses à mão, ou mesmo pegar da Internet já no formato `bibtex`, e depois compilar várias vezes para ver o resultado, muitas vezes (na maioria das vezes) não vale a pena. Mas se você quiser se aventurar,

é só acessar: http://en.wikibooks.org/wiki/TeX/Bibliography_Management
ou a documentação do ABNTex2.

Mas como então formatar a bibliografia? Primeiro, siga a regra da sua instituição ou da revista para a qual você está escrevendo. Uma interpretação mais ou menos correta da ABNT ou da publicação serve, eu aconselho aprender intuitivamente. Depois, crie um capítulo (ou seção, se for artigo). Na classe book ou report:

```
\chapter*{Bibliografia}
\addcontentsline{toc}{chapter}{Bibliografia}
\setlength{\parskip}{11pt}
\setlength{\parindent}{0pt}
```

Na classe article:

```
\section*{Bibliografia}
\addcontentsline{toc}{section}{Bibliografia}
\setlength{\parskip}{11pt}
\setlength{\parindent}{0pt}
```

Ou nas classes KOMA-Script (scrreprt, scrbook) é mais fácil:

```
\addchap{Bibliografia}
\setlength{\parskip}{11pt}
\setlength{\parindent}{0pt}
```

Tabelas

É fácil criar tabelas simples em L^AT_EX, mas tabelas mais complicadas podem ser difíceis. Mas você pode converter as tabelas do Excel ou OpenOffice com plugins: <http://calc2latex.sourceforge.net/> para OpenOffice/Libreoffice e <http://www.ctan.org/tex-archive/support/excel2latex/> para Excel.

Há dois ambientes usados em tabelas: `tabular` e `table`. Geralmente têm a seguinte forma:

texto	texto	texto
texto	texto	texto

Tabela 3: Exemplo de tabela

que foi criada com o seguinte código:

```
\begin{table}[h!]
\centering
\caption{Título da tabela}
\label{exemplotabela}
\begin{tabular}{llp{3cm}}
\hline
texto & texto & texto\\
\hline
\end{tabular}
\end{table}
```

O especificador [h!] obriga a colocar a tabela exatamente “aqui” (h=here). Ela também pode ficar em baixo, no fim da página (b=bottom) ou no centro (c) da página. Se você não explicitar, o L^AT_EX fará um cálculo para inserir a tabela onde ficar melhor, tipograficamente. O acento de exclamação serve para passar por cima dos cálculos e colocar a tabela *exatamente* onde você quer. Os comandos utilizados para tabelas encontram-se na página 28.

Tabelas longas

Para tabelas muito longas (de mais de uma página), podemos usar o ambiente/pacote `longtable`, ao invés de `table`:¹

```
\begin{longtable}{opções}
.....
\end{longtable}
```

¹ Notas de rodapé funcionam com `longtable`, mas não com `table`.

Tabela 4: Comandos utilizados para tabelas

l	coluna justificada à esquerda
c	coluna centralizada
r	coluna justificada à direita
p{largura}	coluna com texto alinhado no topo, com largura especificada (em centímetros ou outra medida)
m{largura}	o mesmo acima, mas alinhado no meio da célula (requer pacote <code>array</code>)
m{largura}	o mesmo acima, mas alinhado no fim da célula (requer pacote <code>array</code>)
	linha vertical
	duas linhas verticais
\hline	linha horizontal
&	separador de colunas
\	começa nova linha (espaço adicional pode ser especificado usando colchetes: por exemplo, [6pt] especifica um espaço de 6 pontos depois da linha)
\newline	começa nova linha dentro da mesma célula
\cline{i-j}	linha parcial horizontal começando na coluna <i>i</i> e terminando na coluna <i>j</i> .

Se você preferir girar a tabela na vertical, use o pacote `rotating` e o ambiente `sidewaystable`:

```
\usepackage{rotating}

\begin{sidewaystable}
\begin{tabular}...
\end{tabular}
\end{sidewaystable}
```

Finalmente, você poderá usar o comando `\listoftables` para gerar uma lista de tabelas (no começo do texto)

Para exemplos mais elaborados (e que dificilmente você utilizará), ver <http://en.wikibooks.org/wiki/LaTeX/Tables>

Formatação

“Formatação” se refere à maioria das coisas que se pode fazer para mudar a aparência do texto: estilo do texto, fonte, tamanho; alinhamento de parágrafos, espaço entre as linhas, indentação, tipos especiais de parágrafo, notas de rodapé, notas de margem etc.

Muitas dessas formatações são exigidas para diferenciar certos elementos do resto do texto. Frequentemente precisamos de enfatizar palavras ou frases. Notas de rodapé são úteis para fornecer informação e explicação extra sem interromper o fluxo do texto principal. Portanto, a formatação é muito importante. No entanto, também é muito fácil abusar dela, e um documento excessivamente rebuscado pode ter uma aparência pior e ser menos legível do que um sem qualquer formatação.

Listas

A formatação conveniente e predizível de listas é uma das muitas vantagens de usar L^AT_EX. Muitos usuários de processadores de texto ficam frequentemente frustrados pelas tentativas desajeitadas do software de perceber quando o usuário quer que as listas começem e terminem. Por outro lado, L^AT_EX lhe dá muito mais estabilidade na geração de listas.

Há basicamente três tipos de listas: itens, enumeração e descrição.

Itemizar

```
\begin{itemize}
\item O primeiro item
\item O segundo item
\item O terceiro item.
\end{itemize}
```

- O primeiro item
- O segundo item
- O terceiro item.

Enumeração

```
\begin{enumerate}
\item O primeiro item
\item O segundo item
\item O terceiro item.
\end{enumerate}
```

1. O primeiro item
2. O segundo item
3. O terceiro item.

Descrição

```
\begin{description}
\item O primeiro item
\item O segundo item
\item O terceiro item.
\end{description}
```

Objeto Descrição do primeiro item
Objeto Descrição do segundo item
Objeto Descrição do terceiro item.

Listas aninhadas (listas dentro de listas)

```
\begin{enumerate}
\item O primeiro item
\begin{enumerate}
\item Primeiro subitem
\item Segundo subitem
\end{enumerate}
\item O segundo item
\item O terceiro item
\end{enumerate}
```

1. O primeiro item
 - (a) Primeiro subitem
 - (b) Segundo subitem
2. O segundo item
3. O terceiro item

Você pode personalizar as listas com o pacote `enumitem`. Assim, para listas como 1.1, 1.2, 1.3:

```
\begin{enumerate}
\item O primeiro item
\begin{enumerate}[label*=\arabic*.]
\item Primeiro subitem
\item Segundo subitem
\begin{enumerate}[label*=\arabic*.]
\item Primeiro sub-subitem
\item Segundo sub-subitem
\end{enumerate}
\end{enumerate}
\end{enumerate}
```

1. O primeiro item
 - 1.1. Primeiro subitem
 - 1.2. Segundo subitem
 - 1.2.1. Primeiro sub-subitem
 - 1.2.2. Segundo sub-subitem

Mas se você só vai usar esse tipo de listas no seu documento, é mais fácil redefinir os comandos:

```
\renewcommand{\labelenumi}{\arabic{enumi}.}
\renewcommand{\labelenumii}{\arabic{enumi}.\arabic{enumii}}
```

Você também pode especificar o tipo de contador que quer: numerais românicos, em maiúsculas, minúsculas ou itálicos. Por exemplo:

```
\begin{enumerate}[label=\Roman*]
\item Primeiro
\item Segundo
\item Terceiro
\end{enumerate}
```

I	Primeiro
II	Segundo
III	Terceiro

As outras opções do pacote `enumitem` (para `\begin{enumerate}[opções]`) são:

`\Alpha* \alpha* \Roman* \roman* \arabic*`

Para soluções mais simples, o pacote `enumerate` é mais intuitivo:

```
\begin{enumerate}[a)]
\item \item \item
\end{enumerate}

a)
b)
c)

\begin{enumerate}[I.]
\item item item
\end{enumerate}

I.
II.
III.
```

No entanto, é bom ter cuidado para só usar *um* dos dois pacotes (`enumerate` ou `enumitem`).

Listas em múltiplas colunas

Usando o pacote `multicol` — `\usepackage{multicol}`—, podemos colocar listas em várias colunas:

```
\begin{multicols}{2}
\begin{enumerate}
\item a
\item b
\item c
\item d
\item e
\item f
\end{enumerate}
\end{multicols}
\end{document}
```

- | | |
|------|------|
| 1. a | 4. d |
| 2. b | 5. e |
| 3. c | 6. f |

Hifenização

O L^AT_EX hifenizará as palavras sempre que necessário. Se houver uma palavra que ele não souber hifenizar, e se ela precisar de hifenização, ela sairá da margem da página no documento final.

Para remediar essa situação, você pode fazer uma lista de hifenizações no começo do seu documento:

```
\hyphenation{lista de palavras}
```

E adicionar as palavras separadas com um hífen.

Uma outra solução é simplesmente hifenizar a palavra recalcitrante com \:-

```
in\-\cons\-\ti\-\tu\-\ci\-\o\-\na\-\lis\-\si\-\ma\-\men\-\te
```

Se você quiser que uma palavra não seja hifenizada, use o comando `\mbox{texto}`, que fará com que a palavra fique sempre junta na mesma linha.

O comando `\fbox{texto}` é semelhante a `\mbox{}`, mas desenha uma caixa ao redor do texto.

Aspas

O L^AT_EX trata aspas da esquerda e da direita como entidades diferentes. Para abrir aspas, escreva a crase (‘) ou (‘‘), e, para fechar, feche com um ou dois apóstrofos (‘ ou ‘‘) ou simplesmente as aspas normais do seu teclado (“”).

Exemplos:

‘Citar’ em Latex	‘Citar’ em Latex	Tabela 5: Aspas
‘‘Citar’’ em Latex	“Citar” em Latex	
‘‘quote’’ in Latex	“Citar” em Latex	
‘‘,quote’’ in Latex	„citar” em Latex	
‘‘Please press the ‘x’ key.’’	“Por favor, pressione a tecla ‘x’.”	
‘‘<<Guillemets>>	« Guillemets » (aspas angulares, utilizadas no português europeu, em francês e alemão, entre outras)	

Geralmente os programas de edição em L^AT_EX como TexStudio e TexWorks têm a opção de mudar as aspas automaticamente, então você geralmente não precisa de se preocupar com isso. Procure no menu de opções do programa.

Para usar estilos de apóstrofes europeus (se não estiver usando o compilador `xelatex`), precisamos de colocar no preâmbulo:

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

Travessão, hífen, meia-risca

Um travessão (—) em L^AT_EX é inserido com: ---.

Um hífen (utilizados para ligar palavras compostas e separar sílabas) é inserido com o hífen normal do teclado.

Uma meia-risca² (–) é inserida com --.

Espaço entre palavras e frases

O L^AT_EX calcula (muito bem, melhor que programas como Word e OpenOffice) os espaços entre as palavras para margens justificadas. Mas ele também coloca um pequeno espaço adicional entre as frases, o que não é muito usual hoje em dia. Para colocar um espaço normal entre as frases, é só adicionar

```
\frenchspacing
```

depois de `\begin{document}`.

Para colocar espaços adicionais entre parágrafos, podem-se usar os seguintes comandos:

² A meia-risca (também chamada de traço de ligação, meio-traço ou traço médio) é um sinal de pontuação que serve para unir os valores extremos de uma série, como números (1–10), letras (A–Z) ou outras, indicando ausência de intervalos na enumeração. Serve igualmente para unir palavras que tenham um nexo lógico (ex.: a viagem Lisboa–Porto). [...] A meia-risca não é o mesmo que o hífen nem que o travessão. (Fonte: <http://pt.wikipedia.org/wiki/Meia-risca>.)

```
\medskip
\bigskip

Ou especificar em centímetros/pontos:

\\[10cm]
```

Enfatizando texto

Para enfatizar uma palavra ou frase, usamos o comando `\emph`:

```
\emph{Enfatizar} uma palavra Enfatizar uma palavra
```

Isso também pode ser feito com os comandos `\textit{}` (“texto em itálico”) e, para negrito, `\textbf{}`.

Estilos de fonte

Os estilos de fonte e seus comandos são explicados na tabela 6.

Comando L ^A T _E X	Equivalente a	Resultado
<code>\textnormal{...}</code>	{\normalfont ... }	a fonte normal do documento
<code>\emph{...}</code>	{\em}	<i>ênfase</i> (geralmente itálico)
<code>\textrm{...}</code>	{\rmfamily}	fonte romana/serifada
<code>\textsf{...}</code>	{\sffamily}	fonte sem serifas
<code>\texttt{...}</code>	{\ttfamily}	fonte com largura fixa
<code>\textit{...}</code>	{\itshape}	<i>itálico</i>
<code>\textbf{...}</code>	{\bfseries}	negrito
<code>\textsc{...}</code>	{\scshape}	VERSALETE (SMALL CAPITALS)

Tabela 6: Estilos de fonte

O sublinhado pode ser feito com `\underline{...}`, mas as palavras com sublinhado não se separam corretamente. Para resolver esse problema, insira

`\usepackage[normalem]{ulem}` no preâmbulo. Com o pacote `ulem`, você pode usar estes comandos: `\uline{...}` (sublinhado) para ênfase; `\uwave{...}` para ondulado e `\sout{...}` para ortado.³ Um ambiente útil para gerar o texto exatamente como foi digitado (como códigos), é o ambiente `verbatim`:

```
\begin{verbatim} ... \end{verbatim}
```

Tudo o que for posto dentro desse ambiente aparecerá no texto final em fonte monoespacada exatamente como está. Para pouco texto, podemos usar o comando

```
\verb+coloque seu texto/comando aqui, entre os ++ ...+
```

Tamanho de fonte

Há alguns comandos predefinidos para tamanhos de fonte. Na verdade, raramente você vai se preocupar com questões como: Qual o tamanho do título? Qual o tamanho do título da seção? Qual o espaço depois do título e antes do texto? Mas se você quiser fazer alguma coisa manualmente, os comandos da tabela 7 são úteis.

Esses comandos afetam *todo* o texto que vai depois deles, então se quiser limitar a uma parte do texto, coloque entre chaves. Exemplo:

```
{\large exemplo de texto} exemplo de texto
```

³ Você pode estar se perguntando: por que essa complicação para sublinhado? A resposta é simples: o sublinhado (assim como o negrito) não é uma prática tipográfica comum. Abra um livro e veja se há algum sublinhado ou negrito. O sublinhado (assim como outras coisas) era usado para ênfase em máquinas de datilografar, e passou para algumas regras de redação acadêmica por isso.

Tabela 7: Tamanho do texto

<code>\tiny</code>	exemplo de texto
<code>\scriptsize</code>	exemplo de texto
<code>\footnotesize</code>	exemplo de texto
<code>\small</code>	exemplo de texto
<code>\normalsize</code>	exemplo de texto
<code>\large</code>	exemplo de texto
<code>\Large</code>	exemplo de texto
<code>\LARGE</code>	exemplo de texto
<code>\huge</code>	exemplo de texto
<code>\Huge</code>	exemplo de texto

Sobrescrito e subscrito

Para sobreescrito (^{0,a,1,2,3}), usamos: `\textsuperscript{}`

Para subscrito (_{0,a,1,2,3}), usamos: `\texsubscript{}` (com o pacote `fixltx2e`).

Para escrever fórmulas químicas, é útil o pacote `mhchem` (<http://www.ctan.org/tex-archive/macros/latex/contrib/mhchem/>). Coloque no preâmbulo:

```
\usepackage[version=3]{mhchem}
```

E escreva suas fórmulas:

Sulfato de amônio é $\text{\ce{(NH4)2SO4}}$. Sulfato de amônio é $(\text{NH}_4)_2\text{SO}_4$.

Símbolos

O L^AT_EX fornece uma imensa quantidade de símbolos, matemáticos ou não. Se você estiver usando um editor de textos como o TexStudio ou Texmaker, você não precisa decorar os comandos, é simplesmente apontar e clicar. Mas alguns pacotes têm que ser colocados no preâmbulo. Os pacotes:

```
\usepackage{textcomp}
\usepackage[official]{eurosym}
\usepackage{amssymb}
\usepackage{wasysym}
```

Vão suprir quase todas as suas necessidades (tabela 8).

Formatação de parágrafos

Os parágrafos em L^AT_EX são geralmente justificados, mas se quiser em outro estilo, pode usar os ambientes/comandos:

Exemplo:

```
\begin{center}
Este é um texto centralizado.
\end{center}
```

Este é um texto centralizado.

Tabela 8: Símbolos comuns

\textdollar	\$
\pounds	£
\textordmasculine	º
\textordfeminine	ª
\dag	†
\ddag	‡
\S	§
\P	¶
\textregistered	®
\officialeuro	€
\textcopyright	©
\male	♂
\female	♀

Tabela 9: Signos astrológicos e do zodíaco

\vernal	♈	\ascnode	♉
\descnode	♉	\fullmoon	○
\newmoon	●	\leftmoon	☽
\rightmoon	☽	\astrosun	☉
\mercury	☿	\venus	♀
\earth	♁	\mars	♂
\jupiter	♃	\saturn	♄
\uranus	♅	\neptune	♆
\pluto	♇	\aries	♈
\taurus	♉	\gemini	♊
\cancer	♋	\leo	♌
\virgo	♍	\libra	♎
\scorpio	♏	\sagittarius	♐
\capricornus	♑	\aquarius	♒
\pisces	♓		

Tabela 10: Alinhamento de parágrafos

Alinhamento	Ambiente	Comando
Justificado à esquerda	flushleft	\raggedright
Justificado à direita	flushright	\raggedleft
Centro	center	\centering

Os parágrafos são indentados (distância entre a margem e a primeira linha do parágrafo) automaticamente. Você pode definir a indentação com o comando:

```
\setlength{\parindent}{1cm}
```

substituindo o número em cm.

Para mudar o espaçamento entrelinhas, use o pacote `\usepackage{setspace}`.

Esse pacote fornece os ambientes:

```
\begin{doublespace}
```

```
...
```

```
\end{doublespace}
```

```
\begin{onehalfspace}
```

```
...
```

```
\end{onehalfspace}
```

```
\begin{singlespace}
```

```
...
```

```
\end{singlespace}
```

E os comandos:

```
\doublespacing
```

```
\onehalfspacing
```

```
\singlespacing
```

Para espaço duplo, um e meio, e um, respectivamente.

Inserção de hiperlinks

Para inserir links que podem ser abertos diretamente no seu navegador ao clicar no pdf, use o pacote

```
\usepackage{hyperref}
```

ou

```
\usepackage{url}
```

e insira o link no comando. Por exemplo:

```
\url{www.wikipedia.org}
```

```
www.wikipedia.org
```

O pacote `hyperref` deve ser colocado como último entre os pacotes do preâmbulo, para evitar incompatibilidades.

Para inserir e-mails clicáveis a partir do pdf, siga este procedimento:

```
\href{mailto:meuemail@wikibooks.org}{meuemail@wikibooks.org}
```

`meuemail@wikibooks.org`

Citando textos

Os ambientes para citação são:

quote para uma citação de geralmente não mais que um parágrafo (parágrafos são separados por linha em branco)

quotation citações longas, geralmente de mais de um parágrafo (parágrafos são indentados)

verse para poesia. Novas linhas com o comando `\|`, e novas estrofes, com uma linha em branco (dois ENTER).

citacao comando do pacote `abntex2` para citações segundo as regras da ABNT.

Notas de rodapé

São inseridas com o comando: `\footnote{...}`.⁴

Cabeçalhos

Há duas maneiras “fáceis” de inserir cabeçalhos, quando necessário: o pacote `fancyhdr` e o pacote `scrpage2` (do pacote KOMA-Script). De qualquer forma, as classes `book` e `scrbook` inserem automaticamente os cabeçalhos, com o número no rodapé.

Se você quiser inserir o número do lado direito no começo da página (uma prática usual por aqui), o jeito mais fácil é pôr `\pagestyle{myheadings}` depois de `\begin{document}` e `\thispagestyle{myheadings}` depois do começo de cada capítulo.

⁴ Neste documento, todas as notas de rodapé são convertidas para notas na margem. Isso pode ser feito com as classes `Tufte-LATEX` ou o pacote `footmisc`.

Modo matemático

O LATEX é geralmente usado por matemáticos ou físicos ou pesquisadores que precisam de muitos símbolos e fórmulas. Mas podemos usar simplesmente porque ele permite um resultado superior sem muito esforço — seja com puro texto, com muitas fórmulas ou muitas figuras. O público-alvo deste “manual” não é esse tipo de usuário, por um simples motivo: o redator (leia-se: eu) não é matemático, não usa fórmulas. Se você precisa de trabalhar com matemática, sugiro dar uma olhada em: <http://en.wikibooks.org/wiki/LaTeX/Mathematics> e/ou <http://mirrors.ctan.org/info/lshort/portuguese-BR/lshortBR.pdf>.

Figuras

Incluindo imagens

Para incluir imagens, usamos o pacote `graphicx` (`\usepackage{graphicx}` no preâmbulo). Além das imagens de formato comum (.jpeg, .png), podem ser incluídas imagens em pdf.

O comando para incluir imagens é:

```
\includegraphics[opções]{nome do arquivo}
```

As opções para inserir imagens são:

`width=xx` Largura=xx. Ex.: `width=5cm`

`height=xx` Altura=xx. Ex.: `height=10cm`

`scale=xx` Escala no fator=xx; por exemplo: `scale=0.5` reduz o tamanho da imagem pela metade, `scale=2` aumenta duas vezes.

`angle=xx` Roda a imagem em xx graus (sentido anti-horário).

`page=x` Se o arquivo de imagem for um pdf com mais de uma página, esse parâmetro possibilita escolher uma outra página que não a primeira.

Para usar mais de uma opção ao mesmo tempo, separe com uma vírgula. A ordem dos comandos pode importar: você deve rodar a imagem (com um ângulo) e *depois* especificar a largura.

Onde guardar as imagens do seu texto

O mais fácil e prático é guardar os arquivos de imagem na mesma pasta do arquivo `.tex`. Se você quiser que o LaTeX procure as imagens em outro lugar, você deve indicar onde elas se encontram com o comando, por exemplo:

```
\graphicspath{~/Pictures} %(geralmente é o diretório de imagens para Linux e Mac)  
\graphicspath{{c:/mypict~1/}} %(geralmente no Windows: My Pictures)
```

Mas você deve ter em mente: não use espaços ou acentos no nome dos arquivos ou das pastas.

Exemplos

```
\includegraphics{coroavray}
```



Figura 2: Gustave Courbet, *Femme à la Vague*, óleo sobre tela, 1868. Metropolitan Museum of Art.

Essa imagem, no entanto, é grande demais e ultrapassa o tamanho da página (por isso nem a exibimos aqui). Se quisermos diminuir o seu tamanho, podemos especificar a escala, a largura e a altura. Assim:

```
\includegraphics[scale=.5]{corotavray}
```



```
\includegraphics[height=10cm]{corotavray}
```



```
\includegraphics[width=.8\textwidth]{corotavray}
```



Neste último caso, especificamos a largura em relação ao texto da página (\textwidth, que poderia ser também a linha, no caso de textos com colunas: \linewidth ou toda a altura do texto: \textheight).

Essa maneira de inserir imagens é prática porque assim você não precisa de ficar se preocupando com medidas.

Imagens como figuras

Há ocasiões em que você pode querer que sua imagem tenha uma legenda e talvez uma referência cruzada. Isso é feito usando o ambiente `figure`. O código mínimo para criar uma figura é:

```
\begin{figure}[opções]
\includegraphics{imagem}
\end{figure}
```

O programa vai calcular onde será melhor colocar a figura para não interromper o fluxo do texto. As opções [] acima podem ser [h], para (aproximadamente) *aqui* (“here”); t, para por a figura no topo da página; b, no fim da página (“bottom”); p, para uma página especial com figuras. Para forçar o programa a colocar exatamente onde você quer, use a exclamação, por exemplo, [h!], “exatamente aqui”.

Você pode usar o comando `\listoffigures` para criar uma lista de figuras (no começo do texto)

O exemplo anterior é bem simples e não tem muitas funções. Geralmente queremos algo como na figura 3, criado com o seguinte código:

```
\begin{figure}[h]
\caption[\textit{Ville d'Avray} de Corot (ca. 1867).]{Jean-Baptiste Corot.
\textit{Ville d'Avray} (ca. 1867).}
Fonte: Wikipedia
(\url{http://fr.wikipedia.org/wiki/Fichier:Corot_villedavray_750pix.jpg})
\label{corot}
\includegraphics[width=.8\textwidth]{corotavray}
\end{figure}
```



Figura 3: Jean-Baptiste Corot. *Ville d'Avray* (ca. 1867). Fonte: Wikipedia (http://fr.wikipedia.org/wiki/Fichier:Corot_villedavray_750pix.jpg)

O comando `\caption{...}` gera o rótulo da figura (ou tabela).

Rótulos ao lado da figura (side captions)

Use o pacote `sidecap`:

```
\usepackage{sidecap}
```

E o ambiente `SCfigure` ao invés de `figure` (figura 4).

```
\begin{SCfigure}
\centering
```



Figura 4: *L'apparition*, de Gustave Moreau (1875).

```
\includegraphics[width=.4\textwidth]{moreau-apparition}
\caption{\textit{L'apparition}, de Gustave Moreau (1875).}
\label{moreau}
\end{SCfigure}
```

Figuras dentro de figuras (subfloats)

Usando o pacote `subfig`, você pode incluir figuras dentro de outras figuras.

As figuras serão colocadas lado a lado; se não houver espaço, vão para a linha inferior. Para colocar uma subfigura manualmente embaixo, é só inserir a marcação de novo parágrafo (\\).

Para inserir espaços entre as figuras, use um dos comandos:

- Espaços brancos, backspaces ou tabs produzirão um espaço regular;
- Espaços matemáticos: \quad, \quad, \;, \,,
- Espaço genérico: \hspace{Xcm};
- Espaço expandido/contraído automaticamente: \hfill

Exemplo:⁵

⁵ [http://en.wikibooks.org/wiki/LaTeX/
FLOATS,_FIGURES_and_CAPTIONS#Subfloats](http://en.wikibooks.org/wiki/LaTeX/FLOATS,_FIGURES_and_CAPTIONS#Subfloats)

```
\usepackage{subfig}
```

```
\begin{figure}
\centering
\subfloat[A gull]{\label{fig:gull}\includegraphics[width=0.3\textwidth]{gull}}
\subfloat[A tiger]{\label{fig:tiger}\includegraphics[width=0.3\textwidth]{tiger}}
\subfloat[A mouse]{\label{fig:mouse}\includegraphics[width=0.3\textwidth]{mouse}}
\caption{Pictures of animals}
\label{fig:animals}
\end{figure}
```



Figure 1: Pictures of animals

Figuras lado-a-lado

Source: <http://theoval.cmp.uea.ac.uk/~nlct/latex/novices/sidebyside.html>

html

The figure environment, should really be called "figures" rather than "figure", as you can have more than one \caption command within the environment, however, since the contents of the figure environment can't have a page break, nor can the figures within a single figure environment float independently of each other, it is more usual to have a separate figure environment for each figure. As a result, people tend to forget that they can have more than one figure in a figure environment, which gives rise to the frequently asked question "how can I have side-by-side figures?"

The answer to this is to put the two figures in the same figure environment. To do this, we can use the minipage environment, which was covered earlier. Recall that the minipage environment creates a horizontal box, which means that two mini-pages can be placed side-by-side on the same line. All you need to do now, is place one image and caption in one mini-page, and the other image and caption in the neighbouring mini-page:

```
\begin{figure}[htbp]
\begin{minipage}{0.5\linewidth}
\centering
\includegraphics{circle}
\caption{A Circle}
\label{fig:circle}
\end{minipage}%
\begin{minipage}{0.5\linewidth}
\centering
\includegraphics{rectangle}
\caption{A Rectangle}
\label{fig:rectangle}
\end{minipage}
\end{figure}
```

Figuras com o tamanho da página em textos de duas colunas

É só colocar um * na frente de figure:

```
\begin{figure*}
\end{figure*}
```

Assim, ela vai ter a largura das duas colunas.

Referências

Referências cruzadas

O L^AT_EX pode inserir referências cruzadas em praticamente qualquer parte do texto. Primeiro, insere-se um rótulo (label):

```
\label{latexbasico}
```

Depois, podemos nos referir ao marcador/rótulo com os comandos:

```
\ref{latexbasico}
```

```
\pageref{latexbasico}
```

O primeiro refere-se à seção, o outro, à página. Assim, inserindo um \label{...} numa parte do texto, em qualquer outra parte podemos nos referir a ela. Veja este exemplo: é a figura 1 na página 16. Eu nem tenho ideia em que página ou seção está, o programa simplesmente vai mostrar no texto final, como neste caso:

A concepção da figura feminina na pintura do simbolista Gustave Moreau (figura 4), pode ser contrastada com a interpretação realista de Courbet (figura 2, p. 40).

Podemos facilmente fazer referências a tabelas, seções, capítulos, figuras etc. E eu posso mudar essa citação de página à vontade, sempre que eu compilar o texto, a paginação e a referência vão aparecer corretas. O único porém é que nem sempre as referências aparecem na primeira vez em que se roda o programa. Simplesmente compile o documento novamente, e se necessário, outra vez.

Hyperlinks

Se quiser hyperlinks clicáveis no seu pdf, use o comando \url{}:

```
\url{www.wikipedia.org} www.wikipedia.org
```


Cores

Para texto ou outros elementos coloridos, use um dos pacotes: `color` ou `xcolor`.

A sintaxe é simples:

```
\textcolor{nome da cor}{Texto nessa cor}
```

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

Tabela 11: As 68 cores da opção `dvipsnames`

Usando a opção `dvipsnames` (`\usepackage[dvipsnames]{xcolor}`), temos 68 opções de cores (ver tabela 11). Para outras opções, ver a documentação do pacote `xcolor`.

Fontes

Para selecionar uma fonte que não é a padrão (Computer Modern), você precisa incluir um pacote de fontes nativo ou usar o pacote `fontspec` e compilar com o comando `xelatex`.

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

No caso, `times` é o pacote da fonte (Times, igual à Times New Roman). Há dezenas desses pacotes, muitos vêm direto na instalação básica do L^AT_EX. Você pode encontrá-las em: <http://www.tug.dk/FontCatalogue/>. Há muitas fontes interessantes, boas e legíveis, e todas elas podem ser utilizadas compilando o documento com o comando: `pdflatex (nome do documento).tex` (ou direto no editor de texto). Por exemplo, temos as fontes: Charter, Palatino, Utopia, Century... Mas se você quiser usar as fontes que vêm com seu computador, você tem que compilar o texto com `xelatex (nome do documento).tex`.

O pacote fontspec

O pacote `fontspec` facilita a escolha de fontes não nativas ao Latex. O arquivo deve ser compilado com o comando `xelatex (nomedoarquivo.tex)`,

```
\usepackage{fontspec}  
\defaultfontfeatures{Ligatures={TeX}}  
\setromanfont{Chaparral Pro}  
\setsansfont{Chaparral Pro}  
\setmonofont[Scale=MatchLowercase]{Consolas}
```

O comando

```
\defaultfontfeatures{Ligatures={TeX}}
```

faz com que conjuntos de caracteres LaTe_X como --- e ‘‘ apareçam como — e “ no texto final.

Depois temos a definição da fonte serifada, sem serifa (geralmente usada para títulos) e monoespaçada:

```
\setromanfont{Chaparral Pro}  
\setsansfont{Chaparral Pro}  
\setmonofont[Scale=MatchLowercase]{Consolas}
```

Claro que você pode (ou deve) substituir o nome das fontes pelas que mais preferir. Com o Adobe Acrobat Reader, vêm as fontes de graça: Minion Pro e Myriad Pro.⁶ Outra fonte interessante é Gentium.

⁶ Para Linux, faça o download de: <http://www.adobe.com/support/downloads/detail.jsp?ftpID=4426>, extraia e instale. Para Windows, vá até a pasta onde instalou o Reader (Arquivos de Programas), procure as fontes e instale.

Idiomas

Para definir a língua em Latex, usa-se o pacote `babel` para compilar em `pdflatex` e o pacote `polyglossia` para compilar em `xelatex`. São esses dois pacotes, em cada caso, que definem as línguas.⁷ A mudança de uma língua para a outra afeta coisas que são automáticas no Latex: abstract, títulos de capítulos, sumário, datas, hifenização, etc. Há uma diferença de preâmbulo, de acordo com o caso.

Para `pdflatex`:

```
\documentclass[12pt,a4paper]{article}
\usepackage[brazil]{babel} %escolha da língua com pacote babel
\usepackage[utf8]{inputenc} %para inserir acentos
\usepackage[T1]{fontenc}
\usepackage{lmodern} %pacote de fonte
```

Pode-se também escolher mais de uma língua, e no caso a última língua será a língua ativa:

```
\usepackage[english,brazil]{package}
```

Nesse caso, escolhe-se a língua ativa com o ambiente

```
\begin{otherlanguage}{english}
...
\end{otherlanguage}
```

sendo que a língua-padrão do documento será a última (português do Brasil).

Com o pacote `polyglossia`:

```
\usepackage{polyglossia}
\setmainlanguage[brazil]
\setotherlanguages[french,english,german,arabic]
\newfontfamily\arabicfont[Script=Arabic]{Adobe Arabic}
```

Podemos fazer isso facilmente com um ambiente:

```
\begin{french}
...
\end{french}
```

ou então simplesmente: `\textfrench{...}`.

A vantagem do pacote `polyglossia` é que é possível inserir mais facilmente textos em várias línguas:

```
\begin{english}
Texto em inglês. \textenglish{Texto em inglês}
\end{english}
```

⁷ Pode-se usar também o pacote `babel` com `xelatex`, mas somente para línguas ocidentais.

Alfabetos não latinos e escritas não ocidentais

Para textos em outros alfabetos ou sistemas de escrita, sem se preocupar com hifenização ou composição de documentos completos, é mais fácil definir a fonte, se a própria fonte que estiver usando não tiver os caracteres. Assim, se estiver escrevendo texto que contenha caracteres gregos, cirílicos etc., use em todo o documento uma fonte que já tenha todos esses caracteres. Então eu posso simplesmente escrever:

Texto em russo:

Лев Николаевич Толстой (28 августа (9 сентября) 1828, Ясная Поляна, Тульская губерния, Российская империя — 7 (20) ноября 1910, станция Астапово, Тамбовская губерния, Российская империя), граф — один из наиболее широко известных русских писателей и мыслителей. (http://ru.wikipedia.org/wiki/Толстой,_Лев_Николаевич)

Texto em grego:

Ο Οδυσσέας Ελύτης (2 Νοεμβρίου 1911 - 18 Μαρτίου 1996), φιλολογικό ψευδώνυμο του Οδυσσέα Αλεπουδέλλη του Παναγιώτη, ήταν ένας από τους σημαντικότερους Έλληνες ποιητές, μέλος της λογοτεχνικής γενιάς του '30.

Se a fonte que eu estiver usando não tiver os caracteres de uma determinada língua, posso definir outra fonte para essa língua usando os comandos:

```
\newfontfamily\russianfont[Script=Cyrillic]{Minion Pro}
\newfontfamily\greekfont[Script=Greek]{Minion Pro}
```

Para escritas não ocidentais também utilizamos o pacote `fontspec` e/ou o pacote `polyglossia`

```
\usepackage{polyglossia}
\setotherlanguage{arabic}
\usepackage{fontspec}
\usepackage{arabxetex}
\newfontfamily\arabicfont[Script=Arabic]{Adobe Arabic}
\begin{arab}
  (texto em árabe)
\end{arab}
```

العربية أكثر لغات المجموعة السامية متعددة

O pacote `arabxetex` usa automaticamente a fonte gratuita Scheherazade, mas, como no exemplo, pode ser especificada qualquer fonte com suporte a árabe/persa/urdu.

Para escrever em chinês, japonês ou coreano, utilizamos o pacote `xeCJK` e o comando `\setCJKmainfont{nome da fonte}`. Uma solução para usar os três sistemas é:⁸

```
\usepackage{xeCJK}
```

⁸ A solução para um texto com os três sistemas (chinês, japonês e coreano) foi copiada de: <http://tex.stackexchange.com/questions/45331/combining-chinese-japanese-and-korean-text-with-xecjk>

```
\setCJKfamilyfont{zhrm}{SimSun}
\setCJKfamilyfont{jarm}{MS Mincho}
\setCJKfamilyfont{korm}{Batang}

\newcommand\Chinese{\renewcommand\CJKrmdefault{zhrm}\CJKnospace\normalfont}
\newcommand\Japanese{\renewcommand\CJKrmdefault{jarm}\CJKnospace\normalfont}
\newcommand\Korean{\renewcommand\CJKrmdefault{korm}\CJKspace\normalfont}
...

```

Outra solução, para textos curtos, é definir a fonte de forma simples:

```
\newfontfamily{\japones}{MS Mincho}
\newfontfamily{\chines}{SimSun}
\newfontfamily{\coreano}{Batang}
```

Colocando entre chaves para o texto definido: {\japones}

おはよう	Ohayo!	“Bom dia!” em japonês
早安	Zao an!	“Bom dia!” em chinês
안녕하세요	An-nyeong-ha-se-yo	“Como vai?”/”Bom dia!” em coreano.

Se você for escrever textos em somente uma dessas línguas, é bem mais simples escolher só uma delas:

```
\setCJKmainfont{Batang} %para coreano
```

Mais informações podem ser obtidas com a documentação dos pacotes `fontspec` e `polyglossia`.

Outros recursos

- Para informações sobre tipografia, ver o livro: *The Elements of Typographic Style*, de Robert Bringhurst.
- Para exemplos de como funciona a tipografia no L^AT_EX, ver: <http://nitens.org/taraborelli/latex>.
- Para modelos de currículos, ver: <http://nitens.org/taraborelli/cvtex>.
- Para sanar suas dúvidas, há um excelente fórum de usuários Latex em: <http://tex.stackexchange.com>

Números de página na borda

```
\setlength{\footskip}{0pt}
\rofoot[\centerpageno{o}]{\centerpageno{o}}
\lefoot[\centerpageno{e}]{\centerpageno{e}}
\def\centerpageno#1{\leavevmode
  \vbox to 0pt{
    \vss
    \hbox to 0pt{\large\normalfont
      \if#1o\kern 2em\else\hss\fi\thepage
      \if#1o\hss\else\kern2em\fi}
    \vskip.5\textheight
    \vskip\baselineskip}}
```

Esses comandos acima são para colocar o número da página na borda.

Também pode colocar: `\fbox{\thepage}` para número de página dentro de quadrado.

Obs: só funciona com classe koma-script e pacote scrpage2.

Caixas de texto

Se você quiser colocar parte do texto dentro de um quadro, a maneira mais fácil é usar o pacote `framed` (`\usepackage{framed}`) e o pacote `xcolor`. O texto dentro do quadro pode ser maior que uma linha, parágrafo ou página, inclusive com cor de fundo. Usando os ambientes `framed` ou `shaded` ou `snugshade`, você pode escrever vários parágrafos dentro da caixa.

Uso:

```
\definecolor{shadecolor}{Gray}{10.0}  
\begin{framed} ... \end{framed}
```

Define a cor da sombra (fundo)

Bordas ao redor do texto

```
\begin{shaded} ... \end{shaded}
```

Texto com sombra de fundo

```
\begin{snugshade} ... \end{snugshade}  
\fcolorbox{Magenta}{Gray}{...}  
\colorbox{Gray}{...}  
\framebox{...}  
\fbox{XXX}
```

Texto com sombra de fundo mais justa.

Define a cor da borda e a cor do fundo.

Um sombreado só do tamanho do texto.

Uma caixa ao redor do texto do tamanho do parágrafo.

XXX

O pacote `mdframed` é bom para fazer caixas coloridas com texto de mais de uma linha ou parágrafo. Você pode especificar diretamente a cor de fundo, e outras características.

Exemplo:

```
\begin{mdframed}[backgroundcolor=lightgray]  
...  
\end{mdframed}
```

I. Quo usque tandem abutere, Catilina,
patientia nostra? quam diu etiam furor
iste tuus nos eludet? quem ad finem
sese effrenata iactabit audacia? Nihilne
te nocturnum praesidium Palati, nihil
urbis vigiliae, nihil timor populi, nihil
concurrus bonorum omnium, nihil hic
munitissimus habendi senatus locus,
nihil horum ora voltusque moverunt?
Patere tua consilia non sentis, constrictam
iam horum omnium scientia teneri
coniurationem tuam non vides? Quid
proxima, quid superiore nocte egeris,
ubi fueris, quos convocaveris, quid
consilii ceperis, quem nostrum ignorare
arbitraris? —Cicero, *Oratio in L. Catilinam Prima*.

Uma entrada de capítulo elaborada

Estilo de capítulo “Veelo”, da classe `memoir`.

```
\usepackage{titlesec}
\titleformat{\chapter}[display]
{\normalfont\Large\raggedleft}
{\MakeUppercase{\chaptertitlename}%
\rlap{\ \resizebox{!}{1.5cm}{\thechapter} \rule{5cm}{1.5cm}}}
{10pt}{\Huge}
\titlespacing*\{\chapter\}{0pt}{30pt}{20pt}
```


Usando Xe \LaTeX e as classes Tufte- \LaTeX

Se você quiser usar um estilo parecido com o dos livros de E. Tufte, as classes `tufte-handout` (para artigos) e `tufte-book` (para livros) são uma opção interessante para sair da mesmice. Mas se quiser compilar com `xelatex` para usar as fontes do sistema, deve usar o código abaixo no preâmbulo para fazer os cabeçalhos das páginas funcionarem direito:

```
\renewcommand{\allcapsspacing}[1]{\addfontfeature{LetterSpace=20.0}\#1}
\renewcommand{\smallcapsspacing}[1]{\addfontfeature{LetterSpace=5.0}\#1}
\renewcommand{\textsc}[1]{\smallcapsspacing{\textsmallcaps{#1}}}
\renewcommand{\smallcaps}[1]{\smallcapsspacing{\scshape\MakeTextLowercase{#1}}}
```


Este texto foi composto e adaptado (uma tradução livre com acréscimos) de: <http://en.wikibooks.org/wiki/TeX/>. Tipologia: Bulmer. A classe utilizada foi *Tufto-L^AT_EX*.



Os direitos desta obra são regidos pela licença **Creative Commons**. Sua distribuição e cópia é livre, desde que citada a fonte, e pode ser modificada, mas não utilizada para fins comerciais. Ver: <http://creativecommons.org/licenses/by-nc-sa/3.0/br/>.