
Curso de Introdução ao L^AT_EX

Prof. Márcio Nascimento da Silva
Universidade Estadual Vale do Acaraú
Coordenação de Matemática

Junho - 2009

Sumário

Introdução	9
1 T_EX e L^AT_EX	11
1.1 O editor de textos Kile	12
1.2 Instalando o L ^A T _E X e o editor Kile	12
1.3 Estrutura de um documento L ^A T _E X	12
1.4 Pacotes	14
1.5 Tipos de Arquivos	15
1.6 Visualização	15
2 Formatação do Texto	17
2.1 Tamanho de letra	17
2.2 Estilo do texto	17
2.3 Fontes	18
2.4 Parágrafo, Espaço e Espaçamento	18
2.4.1 Parágrafo	18
2.4.2 Quebra de linha	18
2.4.3 Espaço Horizontal	19
2.4.4 Espaço Vertical	19
2.4.5 Espaçamento	19
2.5 Alinhamento	19
2.6 Cores	19
2.7 Numeração de Páginas	20
2.8 Notas de Rodapé	20
2.9 Ambiente Verbatim	21
2.10 Colunas	21
3 Inserindo fórmulas matemáticas	23
3.1 Conhecimento básico para fórmulas matemáticas	25
3.1.1 Frações	25
3.1.2 Índices, Expoentes e Raízes	25
3.1.3 Números Binomiais	26
3.1.4 Conjuntos Numéricos	26
3.1.5 Parêntesis, Colchetes e Chaves	26
3.1.6 Módulo e Norma	27
3.1.7 Maior, menor, igual, diferente	27
3.1.8 Fórmulas Trigonométricas	27
3.1.9 Logaritmo e Exponencial	28
3.1.10 Limites, Derivadas e Integrais	28
3.1.11 Somatórios, produtórios	29
3.1.12 União, Interseção, Produto cartesiano	30
3.1.13 Vetores e Conjugados	30
3.1.14 Produto interno, produto vetorial	31
3.1.15 Matrizes	31
3.1.16 Determinantes	32

3.1.17	Sistemas Lineares	33
3.2	Preparando uma lista de exercícios	33
3.2.1	Ambientes de Enumeração	36
3.3	Teoremas, Definições, Proposições	37
3.4	Tabelas	38
3.4.1	Ambiente Table	39
4	Inserindo Imagens	41
4.1	Inserindo figuras lado a lado	42
4.2	Transformação de figuras para o formato EPS	43
4.2.1	Gráficos do GnuPlot	43
4.3	XFig	43
4.4	Outros programas	43
4.5	Lista de Figuras	44
5	Definindo comandos	45
6	Fazendo um livro	47
6.1	Formatando o preâmbulo	47
6.2	Divisão do arquivo \LaTeX em vários arquivos	48
6.3	Capa	48
6.4	Partes	49
6.5	Capítulos	49
6.6	Seções	50
6.6.1	Subseções	51
6.7	Sumário	52
6.8	Lista de Figuras	52
6.9	Referências Bibliográficas	52
6.10	Índice Remissivo	53
	Referências Bibliográficas	54

Lista de Figuras

1.1	Idéia básica de um processador de textos.	11
4.1	Gráfico da função $f(x) = x^2$	42
4.2	Figura da esquerda	43
4.3	Figura da direita	43

Lista de Tabelas

2.1	Tamanhos das letras	18
2.2	Comandos para alinhamento de texto.	20
3.1	Exemplo de tabela com legenda.	40

Introdução

Este manual foi elaborado inicialmente para um curso de Extensão direcionado aos alunos do curso de Matemática da Universidade Estadual Vale do Acaraú em março de 2009. Durante o próprio curso, algumas idéias novas e sugestões apareceram e, na medida do possível, foram acrescentadas a este trabalho. Em junho de 2009 se realiza a segunda edição do curso dessa vez com a participação de professores da rede pública de ensino e professores do próprio curso de Matemática. O objetivo deste curso é apenas iniciar os alunos no \LaTeX dando condições para que possam com sua curiosidade reproduzir tudo que desejem em seus trabalhos. Este material está voltado para a produção de textos na área de ciências exatas, sobretudo matemática. Os exemplos foram opções usadas várias vezes no lugar das definições, na intenção de se obter um melhor entendimento.

Obviamente, este material sempre precisa de melhoras. Contamos com o apoio dos leitores para a melhoria e correção de erros que possam aparecer¹.

Vale registrar o apoio dos alunos Adriano Ximenes, Aprígio Teles² e Thaís Martins, que por já terem um pouco mais de experiência com o \LaTeX se propuseram a ser monitores voluntários neste curso. Espera-se que deste curso tenhamos outros monitores voluntários para projetos futuros.

¹mharcus@gmail.com

²Por questões trabalhistas não pôde participar da segunda edição

Capítulo 1

T_EX e L^AT_EX

T_EX e L^AT_EX são processadores de textos, isto é, interpretam certos comandos e transformam em textos que podem ser, de fato, lidos e entendidos.



Figura 1.1: Idéia básica de um processador de textos.

Os processadores de textos se dividem basicamente em duas categorias:

Editores Visuais: a medida que se vai digitando, o texto já aparece na forma que vai ser impresso.

Editores Lógicos: o processamento é feito em duas etapas. Primeiro, os comandos são digitados num *editor de textos* (por exemplo, **Kile**), é o que chamamos de *arquivo fonte*. Depois esses comandos são compilados gerando um *arquivo de saída* (um arquivo tipo pdf, por exemplo).

O T_EX¹ foi criado no final dos anos 70 por *Donald Knuth* da Stanford University. O objetivo era melhorar a impressão de textos científicos, sobretudo as fórmulas matemáticas. O nome corresponde as primeiras letras da palavra *tecnologia* em grego (τεχ), sendo que o X que aparece na verdade é a letra grega χ (qui).

O L^AT_EX (que se pronuncia “lei-tec” ou “lah-tech”) é um conjunto de comandos que permite usar o T_EX com maior facilidade. Foi criado por *Leslie Lamport* na década de 80.

Hoje muitas universidades (sobretudo americanas e européias) exigem que seus alunos usem T_EX ou L^AT_EX para apresentar suas teses, a fim de garantir que as fórmulas sejam representadas com exatidão ou que a qualidade tipográfica seja uniforme. Muitas publicações científicas também o preferem para a submissão de originais. Vantagens:

- Formatação de qualidade;
- Liberta o autor para concentrar-se no conteúdo em vez da forma;
- Facilidade para trabalhar com fórmulas matemáticas;
- Facilidade para trabalhar com bibliografias e citações;
- Geração automática e sempre correta de sumários, listas de tabelas, listas de figuras, etc.
- Facilidade para criação de índices remissivos;

¹Três primeiras letras da palavra tecnologia em grego: τεχνολογια

- Gerenciamento inteligente de notas de rodapé;
- Exportação para HTML através de programas como latex2html e tex4ht e para RTF (latex2rtf);
- Inserção automática de numeração sequencial de seções;
- Gerenciamento fácil de diferentes estilos de página para capa, folha de rosto, índices, parte pré-textual, parte textual, páginas iniciais de capítulo, apêndice, etc.
- Gerenciamento fácil de documentos a ser impressos em ambos os lados do papel.

Grande problema: Centenas de comandos!

1.1 O editor de textos Kile

É um ambiente de desenvolvendo integrado ao $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. Ele permite que sejam usadas todas as funcionalidades do $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ em modo gráfico, facilitando e customizando o acesso a todos os programas do $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ como compiladores, postprocessing, debug, conversões e ferramentas de visualização; ainda possui vários wizards, interfaces com GnuPlot e XFig, e gerenciamento de projetos. E o principal: vários “botões” e “menus” que evitam a memorização ou gasto de tempo com a pesquisa de boa parte das centenas de comandos necessários para a produção de um texto usando $\text{L}_\text{A}\text{T}_\text{E}\text{X}$. O Kile também avisa sobre eventuais erros de digitação, informando, inclusive, a linha na qual se encontra o erro.

Um matemático francês chamado Pascal Brachet, professor de uma escola secundária francesa, criou o programa **Texmaker**, que possui basicamente os mesmos recursos das IDEs $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ encontradas no mundo Windows (normalmente proprietárias e com custo elevado). O programa mudou de nome mais tarde para **Kile** e acabou sendo passado para outro mantenedor. O programa é feito com a biblioteca Qt e o Kile é uma versão mais adaptada (e dependente) do ambiente KDE. O Texmaker continua existindo, mas como uma versão (bem semelhante e feita com a mesma biblioteca Qt) independente do KDE. Isso tornou possível a existência até de uma versão para MacOS X.

1.2 Instalando o $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ e o editor Kile

1.3 Estrutura de um documento $\text{L}_\text{A}\text{T}_\text{E}\text{X}$

Vamos agora começar a produzir textos mais elegantes que nos editores de textos “da moda”. Um documento $\text{L}_\text{A}\text{T}_\text{E}\text{X}$ se divide basicamente em duas partes:

- Preâmbulo;
- Texto;

No preâmbulo fazemos as definições do documento: margens, estilo, tipo de letra, tamanho de letra, numeração de páginas e etc. Isso tudo é feito através da inserção de *pacotes* (os *packages*). A segunda parte é o texto propriamente dito. Veja abaixo um exemplo da estrutura básica de um documento $\text{L}_\text{A}\text{T}_\text{E}\text{X}$:

```
%-----PRE MBULO-----
\documentclass[a4paper, 12pt]{report}
\usepackage[brazil]{babel}
%-----

%-----TEXTO-----
\begin{document}
Aqui ser , em breve, um texto. Quem sabe sobre matem tica...
\end{document}
```

Na parte do documento onde fica o texto propriamente dito, não há segredos quanto à escrita. Não é preciso se preocupar com mudanças de linhas nem acentos. O alinhamento padrão é o justificado. Caso queira mudar de linha, basta usar o comando

```
\newline
```

Se necessário, pode ser feita até mesmo uma mudança de página:

```
\newpage
```

Todo documento deve ter no seu início o comando

```
\documentclass[op es]{estilo}
```

No campo `op es`, podemos definir tipo de papel e tamanho de letra, por exemplo. No campo `estilo` definimos o tipo de documento que queremos: livro, artigo, carta, etc. Abaixo algumas alternativas para o campo `op es`:

- Tipo de Papel: `a4paper`, `letterpaper`, `a5paper`, `b5paper`, `executivepaper` e `legalpaper`. Caso não seja informado, o formato usado será o `letterpaper`.
- Tamanho da letra: `10pt`, `11pt` ou `12pt`. Caso não seja informado, o tamanho usado é `10pt`.
- Layout da página: `landscape` (paisagem), `twocolumn` (duas colunas), `oneside` (imprime em apenas um lado da página), `twoside` (impressão nos dois lados do papel).
- `titlepage`: para que seja gerada uma página exclusiva para o título. Uma capa.
- `leqno`: numeração das páginas feita no lado esquerdo.
- `openany`: faz com que os capítulos se iniciem em qualquer página, ímpar ou par.

Por exemplo

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

O texto terá duas colunas em papel a4 e letra de tamanho 12pt. Abaixo algumas opções de estilo:

- `article`: artigos em jornais científicos, pequenos relatórios;
- `report`: relatórios mais longos, teses;
- `book`: livros;
- `slides`: Para slides. Usa letras grandes. É necessário o pacote `foiltex`;
- `proc`: Para atas baseadas na classe artigo.
- `minimal`: é a menor classe. Usa apenas um tamanho de letra e de papel. Usada geralmente quando se quer detectar problemas no documento \LaTeX ;
- `letter`: cartas.

Para o propósito de produção de textos matemáticos (listas de exercícios, livros, artigos, etc.) o pré-âmbulo abaixo é bastante satisfatório:

```
%-----PR _ MBULO-----
\documentclass[dvipdfm, a4paper, 12pt]{book} %book, article ou report
\usepackage[brazil]{babel} %linguagem do documento
\usepackage[utf8]{inputenc} %reconhece acento e cedilha
\usepackage{amssymb, amsmath, pxfonts} %permite simbolos matem ticos
\usepackage{mathrsfs} %perimite uso de fontes para conjuntos
\usepackage[normalem]{ulem} %permite sublinhar palavras
\usepackage{mathrsfs} %permite o uso de letras trabalhadas
```

```

\usepackage[top=3cm,left=3cm,right=2cm,bottom=3cm]{geometry} %margens
\usepackage{graphicx} %permite inserir figuras
\usepackage[usenames]{color} %permite letras coloridas
\usepackage{makeidx} %pra criar ndice remissivo
\makeindex %constru o do ndice
%-----
\begin{document}

\pagestyle{headings} %estilo de numera o

\end{document}

```

Observe que inserimos uma linha com vários tracinhos mas iniciada por %. Sempre que esse caractere é digitado, qualquer outra coisa que vier em seguida não é processada. Temos, então, uma forma de fazer comentários dentro do texto. Essa é uma maneira de facilitar correções e mudanças posteriores dentro do nosso texto, uma vez que organiza o documento.

1.4 Pacotes

Os *pacotes* ou *packages* são arquivos que implementam características adicionais para os documentos escritos em \LaTeX . Se usarmos a estrutura básica

```

\documentclass{report}
\begin{document}

\end{document}

```

não podemos, por exemplo, escrever palavras com acentos. Para que isso seja possível, inserimos (no preâmbulo) o pacote que reconhece a acentuação, o pacote

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

Para inserir um pacote, digitamos:

```
\usepackage[op es]{pacote}
```

Abaixo, os pacotes mais utilizados na produção de textos matemáticos:

- **babel**: A linguagem empregada nos títulos. Por exemplo: Capítulo, Sumário, Índice Remissivo. Preenchendo o campo das opções com `[brazil]`, todos os títulos são escritos em português. Se nada for feito, a linguagem usada é o inglês.
- **geometry**: pacote usado para definir as margens. No campo das opções definimos as margens superior, esquerda, direita e inferior usando os comandos `[top=3cm,left=3cm,right=2cm,bottom=3cm]`.
- **graphicx**: pacote que permite a inserção de gráficos e figuras.
- **makeidx**: permite a criação do índice remissivo.
- **amsfonts**, **amssymb**: definem alguns estilos de letras e símbolos para o ambiente matemático. É um conjunto de fontes da *American Mathematical Society*.
- **color**: permite texto colorido
- **inputenc**: permite o uso de caracteres com *til* e *cedilha* sem necessidades de construções do tipo `\c{c}\~ao` (que produz *ção*).

1.5 Tipos de Arquivos

Para se obter uma bela página de um livro, por exemplo, alguns tipos de arquivos são criados. O primeiro deles é o *arquivo fonte* no qual digitamos o texto, cujo formato é `.tex`. O arquivo com esta extensão será processado pelo \LaTeX que gerará um arquivo `.dvi` (Device Independent file). Outros formatos:

- `.log`, `.aux`: guardam informações sobre a última vez que o \LaTeX processou seu documento;
- `.toc`: guarda os cabeçalhos. É usado para produzir o sumário.
- `.lof`: É usado para produzir a lista de figuras;
- `.lot`: É usado para produzir a lista de tabelas;
- `.idx`: guarda todas as palavras que vão para o índice remissivo;
- `.ind`, `.ilg`: usados para produzir o índice remissivo.

São gerados ainda arquivos no formato `.pdf` e `.ps`.

1.6 Visualização

Para a produção de um texto usando \LaTeX são necessárias duas etapas:

- Criação do arquivo fonte;
- Compilação (geração do arquivo de saída);

Um “terceiro passo” seria a visualização deste arquivo. No editor Kile, a criação do arquivo fonte se dá na digitação dos códigos \TeX / \LaTeX que serão salvos num arquivo com extensão `.tex`. Na barra de ferramentas há um botão para compilação do arquivo fonte². Há ainda a opção via barra de menu (COMPILAR » COMPILAR » LATEX) ou, simplesmente, com a combinação das teclas

`alt + F2`

Nesse momento é gerado um arquivo com extensão `.dvi` no qual já podemos ver o resultado dos comandos digitados no arquivo fonte. Para visualizar tal arquivo, basta usar a combinação de teclas

`alt + F3`

Há também a opção na barra de ferramentas e na barra de menu.

²Um botão com uma espécie de engrenagem, catraca.

Capítulo 2

Formatação do Texto

Vejamos agora como formatar um texto.

2.1 Tamanho de letra

Já vimos que no preâmbulo podemos escolher o tamanho da letra entre 10pt, 11pt, 12pt. Alguns comandos podem ser usados para alterar o tamanho das letras em uma determinada parte do texto:

- `{\tiny texto}` produz: texto;
- `{\scriptsize texto}` produz: texto;
- `{\footnotesize texto}` produz: texto;
- `{\small texto}` produz: texto;
- `{\normalsize texto}` produz: texto;
- `{\large texto}` produz: texto;
- `{\Large texto}` produz: **texto**;
- `{\LARGE texto}` produz: **texto**;
- `{\huge texto}` produz: **texto**;
- `{\Huge texto}` produz: **texto**;

A alteração (local) no tamanho do texto depende do tamanho de letra fixado no preâmbulo. Abaixo os tamanhos de acordo com o que é definido no preâmbulo:

2.2 Estilo do texto

Podemos formatar o texto para que ele fique:

- *itálico*: `{\it texto}` ou `textit{texto}`
- **negrito**: `{\bf texto}` ou `textbf{texto}`
- subscrito: `\underline{texto}`

Para os comandos `textit{texto}`, `textbf{texto}` e `\underline{texto}` existem botões na barra de ferramentas do Kile.

Tamanho (preâmbulo)	10pt	11pt	12pt
<code>\tiny</code>	5pt	6pt	6pt
<code>\scriptsize</code>	7pt	8pt	8pt
<code>\footnotesize</code>	8pt	9pt	10pt
<code>\small</code>	9pt	10pt	11pt
<code>\normalsize</code>	10pt	11pt	12pt
<code>\large</code>	12pt	12pt	14pt
<code>\Large</code>	14pt	14pt	17pt
<code>\LARGE</code>	17pt	17pt	20pt
<code>\huge</code>	20pt	20pt	25pt
<code>\Huge</code>	25pt	25pt	25pt

Tabela 2.1: Tamanhos das letras

2.3 Fontes

Basicamente o \LaTeX usa três fontes:

- Esta fonte;
- Esta fonte;
- Esta Fonte.

O primeiro tipo não necessita de comandos. Basta digitar o texto. A segunda é tipo m quina de escrever e a terceira é a fonte Sans Serif ou serifada. O \LaTeX permite o uso de outras fontes mediante adição de pacotes adequados. Por exemplo, se anexamos os pacotes `txfonts` ou `pxfonts`, a fonte é alterada em todo o documento¹.

2.4 Parágrafo, Espaço e Espaçamento

2.4.1 Parágrafo

O parágrafo é feito simplesmente deixando uma linha em branco entre dois parágrafos. Também pode ser feito utilizando o comando `\quad` ou `\qqquad` para um espaçamento um pouco maior.

2.4.2 Quebra de linha

A mudança de linha pode ser feita de várias formas:

- Linha em branco entre duas linhas de texto;
- Duas barras no final da linha: `\\`; Nesse comando ainda podemos colocar o parâmetro de tamanho: `\\[tamanho]`. Aí colocamos, por exemplo, em centímetros o espaçamento que queremos. Se quisermos um espaçamento menor que uma linha, basta colocar o parâmetro com sinal negativo. Por exemplo: `\\[-0.2cm]`.
- Usando o comando `\newline` ao final da linha;
- Usando o comando `\linebreak` ao final da linha. A diferença para o comando `\newline` é que aqui o parágrafo fica justificado.

¹Este documento usa o pacote `pxfonts`.

2.4.3 Espaço Horizontal

Para pequenos espaços na horizontal podemos usar uma barra \. Isso dá um espaço de um caractere. Para espaços um pouco maiores podemos usar \quad ou \qqquad. Se quisermos definir nossos próprios espaços, usamos o comando

$$\backslash\hspace{XXcm}$$

onde XX é o valor do espaço (em centímetros).

2.4.4 Espaço Vertical

Para espaços maiores que um parágrafo podemos usar um comando semelhante ao usado para espaçamento livre horizontal:

$$\backslash\vspace{XXcm}$$

2.4.5 Espaçamento

O espaçamento do texto pode ser definido através do comando \linespread{fator} colocado no pré-âmbulo do documento fonte. Note que esse comando adota o mesmo espaçamento para todo o texto. Se quisermos alterar o espaçamento apenas localmente, usamos o comando

$$\backslash\setlength{\baselineskip}{X.Y\baselineskip}$$

Veja o exemplo a seguir

Salve o Corinthians!!

O campeão dos Campeões!!

Eternamente dentro dos nossos corações!

No documento fonte, temos:

$$\{\backslash\setlength{\baselineskip}{2.5\baselineskip}$$

$$\backslash\textit{Salve o Corinthians!!}$$

$$\backslash\textit{O campeão dos Campeões!!}$$

$$\backslash\textit{Eternamente dentro dos nossos corações!}\backslash\par\}$$

2.5 Alinhamento

Assim como na maioria dos editores de texto, podemos alinhar o texto a esquerda, a direita, justificado ou centralizado. O padrão do \LaTeX é o justificado. Os comandos encontram-se a seguir:

2.6 Cores

É necessário o uso do pacote color no pré-âmbulo da seguinte forma

$$\backslash\usepackage[usenames]{color}$$

desta forma podemos escolher as cores pelos seus nomes. Dois comandos nos permitem mudar a cor de uma determinada parte do texto:

- $\backslash\textcolor{cor}{texto}$

Ação	Comando
Centralizar	<code>\begin{center}</code> TEXTO AQUI! <code>\end{center}</code>
Esquerda	<code>\begin{flushleft}</code> TEXTO AQUI! <code>\end{flushleft}</code>
Direita	<code>\begin{flushright}</code> TEXTO AQUI! <code>\end{flushright}</code>

Tabela 2.2: Comandos para alinhamento de texto.

- `{\color{cor}texto}`

Lembrando que o nome das cor deve estar em inglês. Exemplos:

- `\textcolor{blue}{Tim o Campe o!!!}` produz: **Timão Campeão!!!**
- `\textcolor{red}{Sobral fria!}` produz: **Sobral é fria!**
- `\textcolor{green}{\LaTeX massa!}` produz: **LaTeX é massa!**
- `{\color{magenta}Beleza fundamental}` produz: **Beleza é fundamental**
- `{\color{cyan}Beleza fundamental}` produz: **Beleza é fundamental**

2.7 Numeração de Páginas

O comando `\pagestyle{op o}` permite definir como será a numeração das páginas.

- `\pagestyle{plain}`: imprime o número da página no centro do rodapé.
- `\pagestyle{headings}`: imprime o nome do capítulo atual juntamente com o número da página no cabeçalho.
- `\pagestyle{empty}`: não imprime o número das páginas.

Caso queira não exibir o número de uma página em especial, usa-se

`\thispagestyle{op o}`.

Ainda há outras opções para a numeração na página, através da inclusão de outros pacotes, mas essa tarefa ficará a cargo do leitor.

2.8 Notas de Rodapé

É muito comum o uso de notas de rodapé² seja qual for a natureza do texto. Incluímos esse tipo de nota através do comando

`\footnote{texto da nota de rodap }`

imediatamente após a palavra sobre a qual se quer fazer o comentário.

²Isto é uma nota de rodapé

2.9 Ambiente Verbatim

Na produção deste texto, por várias vezes se faz necessário ver o que se está escrevendo no arquivo fonte. Por exemplo, foi dito acima que para mudar a cor de uma palavra para verde, devemos digitar o seguinte comando

```
\textcolor{green}{Palavra verde}
```

No arquivo fonte, se digitarmos o comando acima e compilarmos, veremos apenas

Palavra verde

Aqui aparece o *ambiente verbatim*, isto é, quando desejamos ler no arquivo de saída o que digitamos, de fato, no arquivo fonte. Existem dois comandos:

```
\verb+ T E X T O +
```

ou

```
\begin{verbatim}
TEXTO
\end{verbatim}
```

2.10 Colunas

O texto pode ser dividido inteiramente em duas colunas ou apenas localmente. Se quisermos que todo o texto tenha duas colunas, usamos como parâmetro do `\documentclass` o comando `twocolumn`:

```
\documentclass[a4,twocolumn,12pt]{book}
```

Se quisermos usar localmente, basta fazer o que se segue:

```
\twocolumn[t tulo]
...
TEXTO
...
\onecolumn
```

A partir do comando `twocolumn[t tulo]` ocorre uma mudança de página e partir daí o texto estará em duas colunas, sendo que o título aparece em uma coluna apenas, no topo da página. O parâmetro título é opcional. O comando `onecolumn` determina o fim das colunas e a partir desse o ponto o texto volta a formatação normal.

Capítulo 3

Inserindo fórmulas matemáticas

As fórmulas matemáticas são digitadas de uma maneira especial. Temos as seguintes opções:

1. Fórmula junto ao texto, na mesma linha:

Uma equação do primeiro grau é uma expressão da forma $ax + b = 0$.

No arquivo fonte, digitamos:

Uma equação do segundo grau é uma expressão da forma $ax^2 + bx + c = 0$.

2. Fórmula centralizada, em uma nova linha:

Uma equação do primeiro grau é uma expressão da forma

$$ax + b = 0$$

No arquivo fonte digitamos,

Uma equação do segundo grau é uma expressão da forma $ax^2 + bx + c = 0$.

3. Fórmulas Numeradas:

Uma equação do segundo grau é uma expressão da forma

```
\begin{equation}
ax^2 + bx + c = 0
\end{equation}
```

Teremos:

Uma equação do segundo grau é uma expressão da forma

$$ax^2 + bx + c = 0 \tag{3.1}$$

Se quisermos escrever uma sequência de equações, podemos usar o ambiente `eqnarray`

Se na equação $3x + y = 2$ fizermos $x = 1$, teremos:

```
\begin{eqnarray}
3x + y = 2 \\
y = 2 - 3x \\
y = 2 - 3 \cdot (1) \\
y = 2 - 3 \\
y = -1
\end{eqnarray}
```

Que resulta em:

Se na equação $3x + y = 2$ fizermos $x = 1$, teremos:

$$3x + y = 2 \quad (3.2)$$

$$y = 2 - 3x \quad (3.3)$$

$$y = 2 - 3.(1) \quad (3.4)$$

$$y = 2 - 3 \quad (3.5)$$

$$y = -1 \quad (3.6)$$

Observe que cada linha ficou numerada. Se não quisermos números em determinadas linhas, usamos o comando `nonumber` ao final da linha:

Se na equação $3x+y=2$ fizermos $x=1$, teremos:

```
\begin{eqnarray}
3x+y=2 \nonumber \\
y=2-3x \\
y=2-3.(1) \\
y=2-3 \nonumber \\
y=-1
\end{eqnarray}
```

Veja que tiramos a numeração de duas linhas. Teremos:

Se na equação $3x + y = 2$ fizermos $x = 1$, teremos:

$$3x + y = 2 \quad (3.7)$$

$$y = 2 - 3x$$

$$y = 2 - 3.(1) \quad (3.8)$$

$$y = 2 - 3$$

$$y = -1 \quad (3.9)$$

Se quisermos tirar a numeração de todas as linhas da expressão, usamos um *asterisco* logo após a palavra `eqnarray` (tanto em `begin` como em `end`):

Se na equação $3x+y=2$ fizermos $x=1$, teremos:

```
\begin{eqnarray*}
3x+y=2 \\
y=2-3x \\
y=2-3.(1) \\
y=2-3 \\
y=-1
\end{eqnarray*}
```

O resultado é:

Se na equação $3x + y = 2$ fizermos $x = 1$, teremos:

$$3x + y = 2$$

$$y = 2 - 3x$$

$$y = 2 - 3.(1)$$

$$y = 2 - 3$$

$$y = -1$$

Podemos ainda melhorar a disposição das equações dentro do ambiente `eqnarray` utilizando o símbolo `&`:

Se na equação $3x+y=2$ fizermos $x=1$, teremos:

```
\begin{eqnarray*}
```



```

3x+y & = & 2\\
  y & = & 2-3x\\
  y & = & 2-3.(1)\\
  y & = & 2-3\\
  y & = & -1
\end{eqnarray*}

```

Obteremos:

Se na equação $3x + y = 2$ fizermos $x = 1$, teremos:

$$\begin{aligned}
 3x + y &= 2 \\
 y &= 2 - 3x \\
 y &= 2 - 3(1) \\
 y &= 2 - 3 \\
 y &= -1
 \end{aligned}$$

3.1 Conhecimento básico para fórmulas matemáticas

Boa parte dos comandos que veremos a seguir aparecem em forma de botão ou menu no editor Kile.

3.1.1 Frações

Há dois comandos para inserir uma fração no texto em \LaTeX :

- $\text{\$}\frac{a}{b}\text{\$} \longrightarrow \frac{a}{b}$
- $\text{\$}\dfrac{a}{b}\text{\$} \longrightarrow \frac{a}{b}$

Perceba que a diferença está no tamanho que cada um ocupa. No primeiro caso, a fração se ajusta ao tamanho da linha. No segundo caso, ela ocupa seu espaço independente do espaçamento na linha. Vejamos:

“A professora começou a falar para seus alunos sobre frações. Iniciou perguntando com quantas laranjas cada aluno da turma de 10 crianças ficaria se ela dividisse as 30 laranjas que tinha igualmente entre eles, isto é, ela desejava efetuar a divisão $\frac{30}{10}$. Os alunos pareciam entender, pois responderam que cada um ficaria com 3 laranjas.”

Observe que a linha na qual a fração aparece teve que ocupar mais espaço. Agora veja o mesmo texto usando o comando $\text{\$}\frac{a}{b}\text{\$}$:

“A professora começou a falar para seus alunos sobre frações. Iniciou perguntando com quantas laranjas cada aluno da turma de 10 crianças ficaria se ela dividisse as 30 laranjas que tinha igualmente entre eles, isto é, ela desejava efetuar a divisão $\frac{30}{10}$. Os alunos pareciam entender, pois responderam que cada um ficaria com 3 laranjas.”

3.1.2 Índices, Expoentes e Raízes

Os índices são definidos usando o caractere _ (*underline*).

$$x_j \longleftarrow \text{\$}x_j\text{\$}$$

Caso o índice tenha mais que um dígito, usamos chaves:

$$x_{15} \longleftarrow \text{\$}x_{\{15\}}\text{\$}$$

Os expoentes são definidos usando o caractere \wedge (acento circunflexo).

$$x^n \leftarrow \$x^n\$$$

Caso se deseje imprimir uma potência com mais de dois dígitos, usamos chaves:

$$x^{20} \leftarrow x^{\{20\}}$$

Eventualmente, precisaremos escrever índices e potências ao mesmo tempo, como por exemplo a_2^{20} . No documento fonte, escrevemos:

$$a_2^{\{20\}}$$

Já a raiz n -ésima de um número, utiliza o comando `\sqrt[ndice]{valor}`. Por exemplo, para imprimir $\sqrt[5]{x}$ escrevemos

$$\sqrt[5]{x} \leftarrow \$\sqrt[5]{x}\$$$

3.1.3 Números Binomiais

Para imprimir o número binomial $\binom{a}{b}$ digitamos no arquivo fonte

$$\binom{a}{b} \leftarrow \$\binom{a}{b}\$$$

Repare que assim como com frações, o número binomial se adequou ao espaçamento da linha. Se desejarmos escrevê-lo em “tamanho natural”, digitamos `\dbinom{a}{b}`.

3.1.4 Conjuntos Numéricos

Para os matemáticos, é muito comum na produção de textos a necessidade de se escrever os símbolos para os conjuntos dos números naturais, inteiros, racionais, reais e complexos. O \LaTeX tem um tipo de fonte que pode ser usada localmente no texto para este fim. É necessário o uso dos pacotes `amsfonts`, `amssymb`, `mathrsfs`.

- $\mathbb{N} \leftarrow \$\mathbb{N}\$$
- $\mathbb{Z} \leftarrow \$\mathbb{Z}\$$
- $\mathbb{Q} \leftarrow \$\mathbb{Q}\$$
- $\mathbb{R} \leftarrow \$\mathbb{R}\$$
- $\mathbb{C} \leftarrow \$\mathbb{C}\$$

Outros tipos de fonte ainda podem ser usadas, por exemplo:

- $\mathcal{ABCDEF} \leftarrow \$\mathcal{ABCDEF}\$$: apenas letras maiúsculas;
- $\mathfrak{ABC}gabc \leftarrow \$\mathfrak{ABC}gabc\$$;

3.1.5 Parêntesis, Colchetes e Chaves

O uso de parênteses e colchetes é feito de maneira simples, basta usar o teclado do computador:

$$(2x^2 - 3x - 1)^2 = 1 \leftarrow \$(2x^2-3x-1)^2=1\$$$

$$[2x^2 - x - 7]^5 = 2x \leftarrow \$[2x^2-x-7]^5=2x\$$$

Já as chaves necessitam de um cuidado a mais, pelo fato de serem usadas para agrupar símbolos, como no caso do expoente e do índice. Para escrever chaves, usamos `\{` e `\}`.

$$\{2x^2 - 3x - 1\}^2 = 1 \leftarrow \$\{2x^2-3x-1\}^2=1\$$$

Um outro problema é que parênteses, colchetes e chaves usados dessa forma têm tamanho fixo.

$$\left(\frac{1}{2}\right)^2 = \frac{1}{4} \leftarrow \text{\texttt{\$}\texttt{(\dfrac{1}{2})^2=\dfrac{1}{4}}\text{\texttt{\$}}}$$

Para corrigir isso, usamos os comandos `\left(` e `\right)`:

$$\left(\frac{1}{2}\right)^2 = \frac{1}{4} \leftarrow \text{\texttt{\$}\texttt{\left(\dfrac{1}{2}\right)^2=\dfrac{1}{4}}\text{\texttt{\$}}}$$

Analogamente, usamos `\left[` e `\right]` para colchetes e `\left\{` e `\right\}` para as chaves.

3.1.6 Módulo e Norma

Para o módulo de um número real ou complexo, podemos usar simplesmente o teclado do computador

$$|x| \leftarrow \text{\texttt{\$}\texttt{|x|}\text{\texttt{\$}}}$$

ou ainda o comando `\vert`.

$$|x| \leftarrow \text{\texttt{\$}\texttt{\vert x \vert}\text{\texttt{\$}}}$$

Para indicar a norma de um vetor, também podemos usar a barra vertical do teclado:

$$\|v\| \leftarrow \text{\texttt{\$}\texttt{\|v\|}\text{\texttt{\$}}}$$

ou ainda o comando `\Vert` (agora o “v” é maiúsculo).

$$\|v\| \leftarrow \text{\texttt{\$}\texttt{\Vert v \Vert}\text{\texttt{\$}}}$$

O problema é que assim como com parênteses, colchetes e chaves, elas têm tamanho fixo. Para corrigir isso, procedemos da mesma maneira que na subseção anterior:

$$\left|\frac{1}{x}\right| \leftarrow \text{\texttt{\$}\texttt{\left|\dfrac{1}{x}\right|}\text{\texttt{\$}}}$$

$$\left\|\frac{2}{3}v\right\| \leftarrow \text{\texttt{\$}\texttt{\left\|\dfrac{2}{3}v\right\|}\text{\texttt{\$}}}$$

3.1.7 Maior, menor, igual, diferente

Os símbolos $>$ (“maior que”), $<$ (“menor que”) e $=$ (“igual a”) podem ser usados diretamente do teclado. Já os símbolos \leq (“menor do que ou igual”), \geq (“maior do que ou igual”) e \neq (“diferente de”) necessitam de comandos especiais:

$$\leq \leftarrow \text{\texttt{\$}\texttt{\leq}\text{\texttt{\$}}}$$

$$\geq \leftarrow \text{\texttt{\$}\texttt{\geq}\text{\texttt{\$}}}$$

$$\neq \leftarrow \text{\texttt{\$}\texttt{\neq}\text{\texttt{\$}}}$$

Todos esses símbolos estão acessíveis na coluna de símbolos que aparecem do lado esquerdo da janela do editor Kile.

3.1.8 Fórmulas Trigonômétricas

Mesmo usando o pacote `babel` as fórmulas trigonométricas estão na notação inglesa, isto é:

$$\text{sen}(x) \longrightarrow \sin(x)$$

$$\text{cos}(x) \longrightarrow \cos(x)$$

$$\text{tg}(x) \longrightarrow \tan(x)$$

$$\text{cotg}(x) \longrightarrow \cot(x)$$

$$\sec(x) \longrightarrow \sec(x)$$

$$\operatorname{cosec}(x) \longrightarrow \csc(x)$$

Podemos corrigir isso usando “novos comandos”. No pré-âmbulo do documento, colocamos o seguinte comando:

$$\backslash\newcommand{\sen}{\rm sen}$$

O primeiro par de colchetes diz o comando que estamos definindo. O segundo par de colchetes contém a função realizada pelo comando. No caso acima, estamos definindo o comando $\backslash\sen$. Ele produzirá

$$\text{sen}$$

Ou ainda

$$\backslash\sen (x) \longrightarrow \text{sen}(x)$$

De maneira análoga definimos as demais funções trigonométricas com a notação portuguesa.

3.1.9 Logaritmo e Exponencial

Exponencial:

$$e^x \longleftarrow \$e^x\$$$

$$\exp(x) \longleftarrow \$\exp(x)\$$$

Logaritmo:

$$\ln y \longleftarrow \$\ln y\$$$

$$\log_a b \longleftarrow \$\log_a b\$$$

3.1.10 Limites, Derivadas e Integrais

Seja $f : \mathbb{R} \longrightarrow \mathbb{R}$ uma função real de uma variável real. A sua *derivada* em um valor $a \in \mathbb{R}$, quando existe, é definida por:

$$\lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} =: f'(a)$$

O comando usado para escrever essa fórmula foi:

$$\lim_{h \rightarrow 0} \frac{f(a+h) - f(a)}{h} =: f'(a)$$

Derivadas de ordem superior:

$$f''(x) \longrightarrow f''(x)$$

$$f'''(x) \longrightarrow f'''(x)$$

$$f^{(n)}(x) \longrightarrow f^{(n)}(x)$$

No caso de derivadas parciais:

$$\frac{\partial f}{\partial x} \longrightarrow \frac{\partial f}{\partial x}$$

Derivadas parciais de ordem superior:

$$\frac{\partial^2 f}{\partial x \partial y} \longrightarrow \frac{\partial^2 f}{\partial x \partial y}$$

$$\frac{\partial^n f}{\partial x^n} \longrightarrow \frac{\partial^n f}{\partial x^n}$$

Gradiente:

$$\nabla f \longleftarrow \$\nabla f\$$$

Laplaciano:

$$\Delta f \leftarrow \Delta f$$

Integral indefinida:

$$\int f(x)dx \leftarrow \int f(x) dx$$

$$\iint f(x)dx \leftarrow \iint f(x) dx$$

$$\iiint f(x)dx \leftarrow \iiint f(x) dx$$

Integral definida:

$$\int_1^{15} f(x)dx \leftarrow \int_1^{15} f(x) dx$$

$$\iint_A f(x)dx \leftarrow \iint_A f(x) dx$$

$$\int_a^b \int_c^d f(x)dx \leftarrow \int_a^b \int_c^d f(x) dx$$

O comando \int quando escrito no meio de um parágrafo se adequa ao tamanho da linha (como no comando $\frac{a}{b}$). Para corrigir isso, usamos

$$\displaystyle \int$$

e temos o seguinte aspecto: \int .

Integrais em caminhos fechados:

$$\oint \langle F(\alpha(t)), \alpha'(t) \rangle dt \leftarrow \oint \langle F(\alpha(t)), \alpha'(t) \rangle dt$$

$$\oint \langle F(\alpha(t)), \alpha'(t) \rangle dt \leftarrow \oint \langle F(\alpha(t)), \alpha'(t) \rangle dt$$

$$\oint \langle F(\alpha(t)), \alpha'(t) \rangle dt \leftarrow \oint \langle F(\alpha(t)), \alpha'(t) \rangle dt$$

$$\oint \langle F(\alpha(t)), \alpha'(t) \rangle dt \leftarrow \oint \langle F(\alpha(t)), \alpha'(t) \rangle dt$$

$$\oint \langle F(\alpha(t)), \alpha'(t) \rangle dt \leftarrow \oint \langle F(\alpha(t)), \alpha'(t) \rangle dt$$

OBSERVAÇÃO: As quatro últimas integrais não “funcionam” se não usarmos o pacote pxfonts

3.1.11 Somatórios, produtórios

Somatório:

$$\sum_{n=1}^{20} n^2 \leftarrow \sum_{n=1}^{20} n^2$$

$$\sum_{n=1}^{\infty} n^2 \leftarrow \sum_{n=1}^{\infty} n^2$$

Assim como no caso dos comandos $\frac{\{\}\{\}}$ e $\dfrac{\{\}\{\}}$ o símbolo de somatório pode ficar pequeno, principalmente se escrito no meio de um parágrafo. Neste caso, podemos usar também o comando $\displaystyle \sum_{n=1}^{\infty} n^2$ para obter $\sum_{n=1}^{\infty} n^2$.

Produtório:

$$\prod_{n=1}^{10} 2^n \leftarrow \backslash prod_{n=1}^{10} 2^n$$

$$\sum_{n=1}^{\infty} 2^n \leftarrow \backslash sum_{n=1}^{\infty} 2^n$$

Caso necessário, podemos usar também o comando

$$\displaystyle \prod_{n=1}^{\infty} 2^n$$

para obter $\prod_{n=1}^{\infty} 2^n$.

3.1.12 União, Interseção, Produto cartesiano

União:

$$A \cup B \leftarrow A \cup B$$

Interseção:

$$A \cap B \leftarrow A \cap B$$

União de uma família de conjuntos:

$$\cup_{i=1}^{\infty} A_i \leftarrow \cup_{i=1}^{\infty} A_i$$

$$\bigcup_{i=1}^{\infty} A_i \leftarrow \bigcup_{i=1}^{\infty} A_i$$

Interseção de uma família de conjuntos:

$$\cap_{i=1}^{\infty} A_i \leftarrow \cap_{i=1}^{\infty} A_i$$

$$\bigcap_{i=1}^{\infty} A_i \leftarrow \bigcap_{i=1}^{\infty} A_i$$

Produto Cartesiano

$$A \times B \times C \leftarrow A \times B \times C$$

3.1.13 Vetores e Conjugados

Elementos em \mathbb{R}^n :

$$X = (x_1, x_2, \dots, x_n) \leftarrow X=(x_1, x_2, \ldots, x_n)$$

Vetores no plano:

$$\vec{v} = x\vec{i} + y\vec{j} \leftarrow \overrightarrow{v}=x\overrightarrow{i} + y\overrightarrow{j}$$

Conjugado de um número complexo:

$$z = a + bi \implies \bar{z} = a - bi \leftarrow z=a+bi \longrightarrow \overline{z}=a-bi$$

3.1.14 Produto interno, produto vetorial

Produto interno:

$$\langle \vec{u}, \vec{v} \rangle \leftarrow \text{\texttt{\$}\texttt{\textbackslash langle \texttt{\textbackslash overrightarrow\{u\}}, \texttt{\textbackslash overrightarrow\{v\}} \texttt{\textbackslash rangle}\$}$$

$$\vec{u} \bullet \vec{v} \leftarrow \text{\texttt{\$}\texttt{\textbackslash overrightarrow\{u\}} \texttt{\textbackslash bullet \texttt{\textbackslash overrightarrow\{v\}} \texttt{\textbackslash \$}}$$

Produto vetorial

$$\vec{u} \times \vec{v} \leftarrow \text{\texttt{\$}\texttt{\textbackslash overrightarrow\{u\}} \texttt{\textbackslash times \texttt{\textbackslash overrightarrow\{v\}} \texttt{\textbackslash \$}}$$

3.1.15 Matrizes

Notação condensada:

$$A = [a_{ij}]_{n \times m} \leftarrow \text{\texttt{\$}A=[a_{\{i\}j}]\texttt{_}\{n \texttt{\textbackslash times} m\}\texttt{\$}}$$

Notação expandida:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}_{3 \times 3}$$

Os comandos que geram essa matriz são

```

 $\$A=\left[ \right.$ 
 $\begin{array}{l} \\ \\ \\ \\ \\ \\ \end{array}$ 
 $a_{\{11\}} \& a_{\{12\}} \& a_{\{13\}} \\ a_{\{21\}} \& a_{\{22\}} \& a_{\{23\}} \\ a_{\{31\}} \& a_{\{32\}} \& a_{\{33\}} \\ \end{array}$ 
 $\right._{\{3 \texttt{\textbackslash times} 3\}}\$$ 

```

Vamos entender o conjunto de comandos. Na parte mais interna, temos o ambiente¹ *array*. Note que logo junto à palavra *array* há o parâmetro `{l1l}`. A quantidade de 1 (éle) (nesse caso 3) indica a quantidade de colunas da matriz. A letra l diz que o alinhamento é à esquerda (left, em inglês). Assim, o parâmetro que define a quantidade de linhas, poderia ter sido definido de uma das seguintes formas:

- `{rrr}`: três colunas, todas alinhadas à direita;
- `{ccc}`: três colunas, todas centralizadas;

Poderíamos, até mesmo, ter colunas com alinhamentos diferentes, por exemplo, definido esse parâmetro por `{rlc}`, que indica alinhamento à direita na primeira coluna, alinhamento à esquerda na segunda coluna e centralização na terceira coluna.

Dentro do *array* temos as linhas da matriz, onde os elementos são separados por `&`. Além disso, a mudança de linha é caracterizada pelas barras `\`.

Matriz, caso geral:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

Comandos:

¹Tem um início e um fim.

```


$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$


```

Matriz quadrada, caso geral:

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}_{n \times n}$$

Comandos:

```


$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}_{n \times n}$$


```

Matriz coluna

$$A = \begin{bmatrix} 10 \\ -12 \\ 8 \end{bmatrix}_{3 \times 1}$$

Comandos:

```


$$A = \begin{bmatrix} 10 \\ -12 \\ 8 \end{bmatrix}_{3 \times 1}$$


```

3.1.16 Determinantes

Dada uma matriz quadrada $A = [a_{ij}]_{n \times n}$ denotamos seu determinante por

$$\det(A) \leftarrow \det(A)$$

ou

$$\begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{vmatrix}$$

Cujos comandos são


```


$$\begin{array}{cccc} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{array}$$


```

3.1.17 Sistemas Lineares

Sistema com m equações e n variáveis:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$

Comandos:

```


$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \end{cases}$$


```

Repare que não há a segunda chave, fechando a expressão, no entanto, devemos informar isso através do comando

`\right.`

Atentai para o “ponto” no final.

3.2 Preparando uma lista de exercícios

Uma das maiores necessidades de um professor é a produção de listas de exercícios e de provas. Usando o \LaTeX , além de um resultado mais elegante, temos maior praticidade.

Cabeçalho

Vamos supor que a lista de exercícios seja para os alunos da Universidade Estadual Vale do Acaraú. Então a estrutura do documento será a seguinte:

```

%-----
PRE-AMBULO
%-----

%-----
\begin{document}
%-----CABE ALHO-----
\textbf{Universidade Estadual Vale do Acaraú}

\textbf{Coordenação de Matemática}

\textbf{1ª Lista de Exercícios - Cálculo Diferencial e Integral I}

\textbf{Professora: Ana Cibely Aragão Monteiro}

```

%-----

%-----QUEST ES-----

%-----

\end{document}

No arquivo de saída, teremos:

Universidade Estadual Vale do Acaraú
Coordenação de Matemática
1ª Lista de Exercícios - Cálculo Diferencial e Integral I
Professora: Ana Cibely Aragão Monteiro

Agora, passemos às questões. Sendo uma lista de exercícios ou prova, é de se esperar que ocorra uma numeração das questões. Para isso usamos o ambiente `enumerate`. Vejamos como funciona. Considere o exemplo:

```
\begin{enumerate}
\item Mostre que toda fun o deriv vel cont nua.
\item Mostre que toda fun o cont nua integr vel.
\item Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

\end{enumerate}
```

Como resultado, temos:

-
1. Mostre que toda função derivável é contínua.
 2. Mostre que toda função contínua é integrável.
 3. Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

Suponha agora que uma questão necessite de subitens. Faremos como se segue

```
\begin{enumerate}
\item Mostre que toda fun o deriv vel cont nua.
\item Mostre que toda fun o cont nua integr vel.
\item Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

\item Calcule o limite:
```

```

\begin{enumerate}
\item 
$$\lim_{x \rightarrow 1} (x-1)$$

\item 
$$\lim_{x \rightarrow 0} x^2$$

\item 
$$\lim_{x \rightarrow 2} e^x$$

\end{enumerate}
```

```
\end{enumerate}
```

Observe que criamos um novo ambiente `enumerate` dentro de outro ambiente também `enumerate`. Como resultado, teremos:

1. Mostre que toda função derivável é contínua.
2. Mostre que toda função contínua é integrável.
3. Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\operatorname{sen} x}{x}$$

4. Calcule o limite:

(a) $\lim_{x \rightarrow 1} (x - 1)$

(b) $\lim_{x \rightarrow 0} x^2$

(c) $\lim_{x \rightarrow 2} e^x$

Uma outra maneira de se inserir subitens é através do comando `\subitem`. Neste caso, devemos digitar também o número ou letra do item. Vejamos:

```
\begin{enumerate}
\item Mostre que toda fun o deriv vel cont nua.
\item Mostre que toda fun o cont nua integr vel.
\item Encontre o limite
$$\lim_{x \to 0} \operatorname{dfrac}{\operatorname{sen} x}{x}$$
\item Calcule o limite:

\subitem (i) $\displaystyle \lim_{x \to 1} (x-1)$
\subitem (ii) $\displaystyle \lim_{x \to 0} x^2$
\subitem (iii) $\displaystyle \lim_{x \to 2} e^x$

\end{enumerate}
```

Como resultado, temos:

1. Mostre que toda função derivável é contínua.
2. Mostre que toda função contínua é integrável.
3. Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\operatorname{sen} x}{x}$$

4. Calcule o limite:

(i) $\lim_{x \rightarrow 1} (x - 1)$

(ii) $\lim_{x \rightarrow 0} x^2$

(iii) $\lim_{x \rightarrow 2} e^x$

É extremamente aconselhável que se use cometários, para facilitar a identificação das questões:

```

\begin{enumerate}
%-----
\item Mostre que toda fun o deriv vel cont nua.
%-----
\item Mostre que toda fun o cont nua integr vel.
%-----
\item Encontre o limite

$$\lim_{x \rightarrow 0} \frac{\sin x}{x}$$

%-----
\item Calcule o limite:

\begin{enumerate}
\item  $\lim_{x \rightarrow 1} (x-1)$ 
\item  $\lim_{x \rightarrow 0} x^2$ 
\item  $\lim_{x \rightarrow 2} e^x$ 
\end{enumerate}

\end{enumerate}

```

3.2.1 Ambientes de Enumeração

Além do ambiente `enumerate` que enumera automaticamente os itens, temos ainda ambientes que apenas descrevem os itens:

Ambiente `itemize`

```

\begin{itemize}
\item Primeira opção;
\item Segunda opção;
\item Terceira opção.
\end{itemize}

```

Como resultado, temos:

-
- Primeira opção;
 - Segunda opção;
 - Terceira opção.
-

Ambiente `description`

```

\begin{description}
\item Primeira opção;
\item Segunda opção;
\item Terceira opção.
\end{description}

```

Como resultado, temos:

Primeira opção;

Segunda opção;

Terceira opção.

O ambiente `description` pode ser usado quando se quer criar a própria enumeração:

```
Axiomas de Adição:
\begin{description}
\item (A1) Associatividade;
\item (A2) Comutatividade;
\item (A3) Existência do elemento neutro;
\item (A4) Elemento simétrico.
\end{description}
```

Como resultado, temos:

Axiomas de Adição:

(A1) Associatividade;

(A2) Comutatividade;

(A3) Existência do elemento neutro;

(A4) Elemento simétrico.

3.3 Teoremas, Definições, Proposições

Para a produção de um texto matemático que vá além de uma lista de exercícios, provavelmente será necessário enunciar algum teorema ou definição.

Ainda no pré-âmbulo do documento, podemos definir ambientes para estes propósitos. Queremos como resultado algo do tipo:

Teorema 3.3.1 (Teorema de Pitágoras) *Sejam a, b catetos de um triângulo retângulo e c sua hipotenusa. Então*

$$c^2 = a^2 + b^2$$

Observe o estilo da letra, numeração da fórmula e título do Teorema. Primeiramente, vamos criar o ambiente «TEOREMA»

```
\newtheorem{teo}{Teorema}[section]
```

O primeiro parâmetro diz que o comando a ser digitado (no texto \LaTeX) é o seguinte:

```
\begin{teo}
Sejam  $a, b$  catetos de um triângulo retângulo e  $c$  sua hipotenusa. Então
 $c^2 = a^2 + b^2$ 
\end{teo}
```

O segundo parâmetro diz com que nome esse ambiente vai aparecer no texto. No caso, Teorema:

Teorema 3.3.2 *Sejam a, b catetos de um triângulo retângulo e c sua hipotenusa. Então*

$$c^2 = a^2 + b^2$$

O terceiro parâmetro diz respeito a numeração empregada nos teoremas que aparecerão no texto. Neste caso, a numeração segue a ordem das seções. Como esta seção é a 3.3, segue que o primeiro teorema desta seção será o 3.3.1, o segundo teorema da seção será o 3.3.2 e assim por diante.

Se quisermos dar um nome ao teorema, basta digitar entre colchetes o nome desejado logo em seguida ao `\begin{teo}`:

```
\begin{teo}[Teorema de Pit goras]
Sejam $a$, $b$ catetos de um tri ngulo ret ngulo e $c$ sua hipotenusa. Ent o
$$c^2=a^2+b^2$$
\end{teo}
```

Criando o ambiente Definição

```
\newtheorem{defin}{Defini o}[section]
```

```
\begin{defin}[Produto Interno]
O produto interno de dois vetores $(x,y,z)$, $(x_1,y_1,z_1)$ do $\mathbb{R}^3$ definido por
$\langle (x,y,z),(x_1,y_1,z_1) \rangle = x.x_1+y.y_1+z.z_1$
\end{defin}
```

Que resulta em

Definição 3.3.1 (Produto Interno) *O produto interno de dois vetores (x, y, z) , (x_1, y_1, z_1) do \mathbb{R}^3 é definido por*

$$\langle (x, y, z), (x_1, y_1, z_1) \rangle = x.x_1 + y.y_1 + z.z_1$$

Para criar os outros ambientes (Proposição, Lema, Corolário, etc.) procede-se da mesma forma.

3.4 Tabelas

Para inserir uma tabela no documento \LaTeX precisamos do ambiente `tabular`:

```
\begin{tabular}{lll}
x & y & z\\
1 & 2 & 3\\
-1 & 4 & 0
\end{tabular}
```

Estes comandos produzem

x	y	z
1	2	3
-1	4	0

Observe que o parâmetro l alinha as colunas à esquerda (left). Além disso, *entre a última linha de texto antes da tabela e a tabela, deve haver uma linha em branco no arquivo fonte*. Coloquemos linhas nessa tabela:

```
\begin{tabular}{lll}
\hline
x & y & z\\
\hline
1 & 2 & 3\\
-1 & 4 & 0\\
\hline
\end{tabular}
```

Daí

x	y	z
1	2	3
-1	4	0

Podemos, ainda, inserir linhas verticais. Fazemos isso nos parâmetros de alinhamento:

```
\begin{tabular}{l|l|l}
\hline
x & y & z\\
\hline
1 & 2 & 3\\
-1 & 4 & 0\\
\hline
\end{tabular}
```

Daí

x	y	z
1	2	3
-1	4	0

Se quisermos fechar a tabela por completo, colocamos as linhas que ainda faltam (horizontais usando o comando `\hline` e as verticais através de barras verticais nos parâmetros de alinhamento):

```
\begin{tabular}{l|l|l|l}
\hline
x & y & z & \\
\hline
1 & 2 & 3 & \\
\hline
-1 & 4 & 0 & \\
\hline
\end{tabular}
```

Daí

x	y	z
1	2	3
-1	4	0

3.4.1 Ambiente Table

Dependendo da necessidade, necessitamos de legenda para as tabelas. Para esse propósito usamos o ambiente `table`. Vejamos o exemplo abaixo

```
%-----IN CIO DE TABELA-----
\begin{table}[h]
\begin{center} %aqui centralizamos a tabela
\begin{tabular}{l|l|l} %-----dividir em duas colunas
\hline
N mero & Nome\\
\hline %----- linha horizontal
01 & Francisco\\
02 & Jos \\
03 & Maria\\
\end{tabular}
\end{center}
\end{table}
```

```

\hline %----- linha horizontal
\end{tabular}%--- fechamento do ambiente tabular
\end{center} %fim da centraliza o da tabela
\caption{Exemplo de tabela com legenda.} %legenda da tabela
\end{table}
%-----FIM DA TABELA-----

```

Esses comandos produzem:

Número	Nome
01	Francisco
02	José
03	Maria

Tabela 3.1: Exemplo de tabela com legenda.

Também podemos criar listas de tabelas através do comando

`\listoftables`

que comumente vem após o sumário, caso se esteja produzindo uma monografia, por exemplo.

Capítulo 4

Inserindo Imagens

Uma parte um tanto quanto delicada é a inserção de figuras num documento \LaTeX . Faz-se necessário:

- Inserção do parâmetro `dvipdfm` em `documentclass`:
`\documentclass[dvipdfm, a4paper, 12pt] {article}`
- Pacote `graphicx` para inclusão de figuras EPS.
- Transformação da figura para o formato EPS¹.

Há um ambiente próprio para inclusão de figuras:

```
\begin{figure}
\includegraphics[op es]{nome_do_arquivo}
\end{figure}
```

Em opções, podemos definir altura, largura, proporção e ângulo de rotação da figura:

- Altura: `height=20pt` ou `height=5cm`, por exemplo;
- Largura: `width=22pt` ou `width=7.5cm`, por exemplo;
- Proporção: Percentual do tamanho da figura. Por exemplo: `scale=0.5` significa que a figura aparecerá num tamanho igual a 50% do tamanho original.
- Ângulo de rotação: Rotação em graus da figura com relação a horizontal, sentido antihorário. Por exemplo: `angle=120`.

O nome do arquivo deve ser digitado com o caminho completo. Por exemplo, suponha que o arquivo tenha nome `figura1.eps` e esteja no diretório `aluno`. Então, o caminho completo é:

`/home/aluno/figura1.eps`

No ambiente `figure` podemos definir ainda outros parâmetros:

- Legenda: através do comando `\caption{texto da legenda}`. As legendas são numeradas automaticamente.
- Marca: através do comando `\label{marca}`. Fora do ambiente `figure`, se for necessário citar tal figura, usamos o comando `\ref{marca}`. A referência é feita automaticamente.
- Posicionamento: através do parâmetro `h`, `t` ou `b` logo em seguida ao

`\begin{figure}`

podemos indicar que a figura deva ser colocada exatamente nesse trecho do texto (`h = "here"`), ou no topo da página em que ela aparecer (`t = "top"`) ou ainda na parte de baixo da página (`b = "bottom"`). Por exemplo:

¹É possível acrescentar figuras em outros formatos, no entanto, transformá-las para EPS torna o processo mais fácil.

```

\begin{figure}[h]
\includegraphics[scale=0.7]{/home/aluno/figuras/grafico1.eps}
\caption{Gráfico da função  $f(x)=x^2$ }
\label{grafico1}
\end{figure}

```

Estes comandos incluem no topo da página, o gráfico da função $f(x) = x^2$.

- Centralizar: através do comando `\centering`.

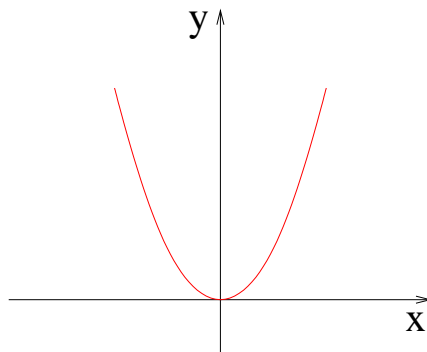


Figura 4.1: Gráfico da função $f(x) = x^2$

4.1 Inserindo figuras lado a lado

Podemos colocar uma ou mais figuras lado a lado, utilizando um único ambiente `figure` e um ambiente `minipage` (minipágina) para cada figura, de modo que a soma das larguras de todas as minipáginas não ultrapasse a largura da linha (`\linewidth`).

O exemplo abaixo mostra as figuras `fun_modular.eps` e `fun_quad.eps` lado a lado com cada uma delas ocupando 45% da linha atual.

```

%-----AMBIENTE DE FIGURA-----
\begin{figure}[h]
\centering
%-----PRIMEIRA FIGURA-----
\begin{minipage}[b]{0.45\linewidth}
\includegraphics[width=\linewidth]{fun_modular.eps}
\caption{Figura da esquerda}
\end{minipage}
%-----
\hfill
%-----SEGUNDA FIGURA-----
\begin{minipage}[b]{0.45\linewidth}
\includegraphics[width=\linewidth]{fun_quad.eps}
\caption{Figura da direita}
\end{minipage}
%-----
\end{figure}
%-----

```

O parâmetro `[b]` alinha as minipáginas por baixo, de modo que as legendas apareçam na mesma linha. Os comandos `\hfil`, `\hfill` e `\hfilneg` servem para definir o espaço entre as minipáginas. Observe o comando `\hfill`² que aparece entre as minipáginas. Não deve haver linhas em branco entre

²ou outro comando para definir o espaço entre as minipáginas

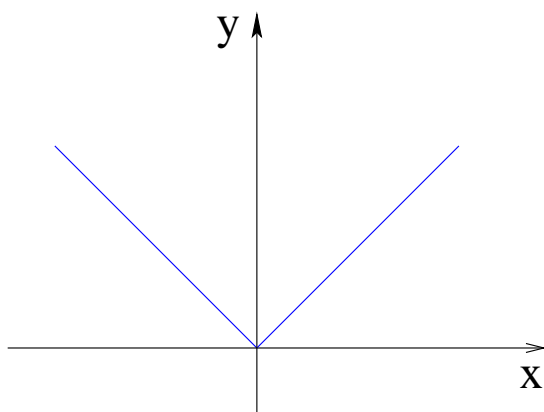


Figura 4.2: Figura da esquerda

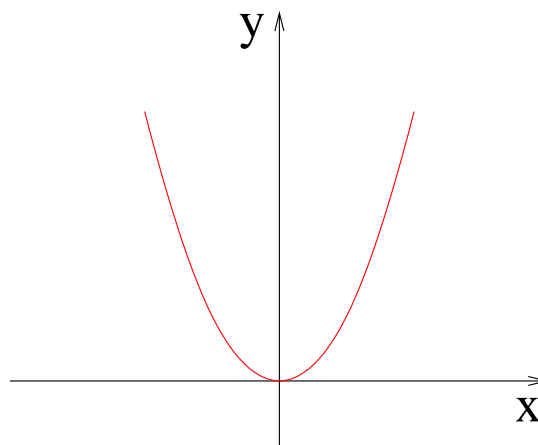


Figura 4.3: Figura da direita

o final da primeira minipágina, o comando e o início da segunda minipágina (a linha que aparece no exemplo acima está comentada!).

4.2 Transformação de figuras para o formato EPS

4.2.1 Gráficos do GnuPlot

Uma maneira bastante prática é a seguinte:

- Gere o gráfico no GnuPlot.
- Capture a janela do gráfico (na versão Linux/Ubuntu ou Debian: APLICAÇÕES » ACESSÓRIOS » CAPTURAR IMAGEM DA TELA)
- Abra figura com algum editor de imagens (GIMP, por exemplo);
- Em ARQUIVO, escolha a opção «EXPORTAR...» e escolha a opção EPS.

Gerando EPS pelo próprio GnuPlot

- `gnuplot> f(x)=x**2-x`
- `gnuplot> set terminal postscript portrait enhanced color`
- `gnuplot> set output "/home/aluno/teste1.eps"`
- `gnuplot> replot`

Agora basta incluir o arquivo `/home/aluno/teste1.eps` no ambiente `figure`.

4.3 XFig

Uma poderosa ferramenta para construção de gráficos, diagramas, figuras e coisas do tipo é o programa XFig. No programa, mais precisamente no menu «FILE» há a opção «EXPORT...». Ai basta digitar o nome do arquivo no campo OUTPUT FILE sem esquecer a extensão `.EPS` e depois clicar em «EXPORT».

4.4 Outros programas

Caso não seja possível usar as opções acima, faz-se necessário um programa que faça desenhos, figuras mas que tenha a opção de exportar no formato `.EPS`.

4.5 Lista de Figuras

Algumas vezes é interessante termos uma lista de figuras, logo após o índice do trabalho, por exemplo. Fazemos isso através do comando

```
\listoffigures
```

A figura que tiver legenda aparecerá nessa lista.

Capítulo 5

Definindo comandos

Muitas vezes na produção de um documento \LaTeX , usamos comandos que se repetem ao longo do texto. Esses comandos podem ser grandes, podendo ser até combinações de comandos, gastando assim mais tempo na digitação. Para melhorar isso, o \LaTeX permite a criação de novos comandos.

Suponha que na digitação de uma monografia, você precisa usar sucessivas vezes o símbolo dos números reais \mathbb{R} . Para que este símbolo apareça na tela, precisamos digitar

```
\mathbb{R}
```

Que tal se pudéssemos digitar apenas o comando abaixo?

```
\R
```

Isso é possível definindo no preâmbulo um novo comando:

```
\newcommand{comando}{a o}
```

No nosso caso, teríamos

```
\newcommand{\R}{\mathbb{R}}
```

e sempre que precisarmos no texto do símbolo dos números reais, basta digitar \mathbb{R} .

Podemos também criar comandos para frases inteiras. Suponha que você está digitando um trabalho para o qual necessite da frase “Ai meu Deus, esse trabalho é muito difícil” em todas as páginas, para chamar a atenção do professor. Você pode criar um comando para esta frase:

```
\newcommand{\apelo}{Ai meu Deus, esse trabalho muito dif cil}
```

Sempre que você digitar `\apelo` esta frase aparecerá no seu arquivo. Por exemplo, se você digitar o seguinte parágrafo:

Ei, professor, `\apelo`, `\apelo`, `\apelo`, `{\it \apelo}`, `{\bf \apelo}`!

Aparecerá:

Ei, professor, Ai meu Deus, esse trabalho é muito difícil, Ai meu Deus, esse trabalho é muito difícil, Ai meu Deus, esse trabalho é muito difícil, <i>Ai meu Deus, esse trabalho é muito difícil</i> , Ai meu Deus, esse trabalho é muito difícil !
--

E se você estiver escrevendo um livro, provavelmente precisará fazer muitos e muitos comandos, concorda? Mas desse jeito seu preâmbulo vai ficar lotado! Podemos, então, criar um lugar exclusivo para os novos comandos: um novo arquivo! Crie um novo arquivo e o chame de `macros.tex`. Memorize bem o diretório no qual você vai guardar esse arquivo. Você vai poder usar para todos os livros que for escrever ao longo de sua carreira. Basta você incluir no arquivo os seguintes comandos:

```
%este o arquivo no qual ficar o meus pr prios comandos
\newcommand{\R}{\mathbb{R}}
\newcommand{\apelo}{Ai meu Deus, esse trabalho muito dif cil}
```

Agora no preâmbulo de seu arquivo \LaTeX basta colocar o comando

```
\input{/home/minhapasta/macros.tex}
```

e você terá um arquivo com todos esses comandos novos que poderão ser usados em qualquer outro arquivo \LaTeX , basta que você coloque o comando acima no preâmbulo de seus documentos. Outra vantagem é que você pode ir aumentando o número de comandos próprios a qualquer tempo, e acessar esses comandos também a qualquer tempo.

Capítulo 6

Fazendo um livro

Neste capítulo veremos como escrever um livro usando \LaTeX . Primeiramente, adiantamos que você vai precisar criar vários arquivos, por isso, antes de tudo, crie uma pasta com o nome do seu “projeto”, por exemplo a pasta MEU LIVRO. Depois, crie o seu documento \LaTeX ao qual vamos nos referir por *principal*. Salve-o com o nome livro.tex.

6.1 Formatando o preâmbulo

No preâmbulo, escolha o estilo book logo na primeira linha.

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

Isso permite que seu documento tenha a estrutura de um livro (sumário, partes, capítulos, seções, etc.). Escolha o tipo de papel e letra que lhe agradar. Depois coloque todos os pacotes que você julgar necessário (muitos deles só se mostram necessários ao longo do trabalho). Aqui vamos dar uma sugestão partindo do princípio que você vá fazer um livro voltado para a Matemática:

```
\documentclass[dvipdfm, a4paper, 12pt]{book}
\usepackage[brazil]{babel} %linguagem do documento
\usepackage[utf8]{inputenc} %reconhece acento e cedilha
\usepackage{amssymb, amsmath, pxfonts} %permite simbolos matematicos
\usepackage{mathrsfs} %permite uso de fontes para conjuntos
\usepackage[normalem]{ulem} %permite sublinhar palavras
\usepackage{mathrsfs} %permite o uso de letras trabalhadas
\usepackage[top=3cm,left=3cm,right=2cm,bottom=3cm]{geometry} %margens
\usepackage{graphicx} %permite inserir figuras
\usepackage[usenames]{color} %permite letras coloridas
\usepackage{makeidx} %pra criar ndice remissivo
\makeindex %constru o do ndice
```

Ainda no preâmbulo, coloque seus arquivos de comandos próprios:

```
\input{/home/minhapasta/macros.tex}
```

Agora podemos passar para a digitação do livro. O texto propriamente dito ficará no ambiente document:

```
\begin{document}
MEU LIVRO
\end{document}
```

Quando você tiver essa experiência, verá que terá que voltar ao texto várias e várias vezes. Se seu livro tiver muitas páginas, digamos, umas 200, teremos muitas e muitas linhas de código no arquivo \LaTeX e isso dificulta o acesso a algum erro que se encontre na página 76, por exemplo. Desta forma, devemos buscar uma melhor organização para o texto. Algo que possibilite um melhor acesso em caso de correções.

6.2 Divisão do arquivo \LaTeX em vários arquivos

Vimos anteriormente que o comando

```
\input{/home/minhapasta/macros.tex}
```

faz com que o documento \LaTeX “enxergue” os comandos que se encontram no seu arquivo `macros.tex`. Um comando que faz algo parecido, mas com textos é o comando:

```
\include{nome_do_arquivo.tex}
```

Dentro do seu documento \LaTeX , no ambiente `{document}`, no momento em que aparece o comando `\include{nome_do_arquivo.tex}` o processamento vai imediatamente buscar o texto que se encontra em tal arquivo. Por exemplo, suponha que no arquivo `capitulo1.tex` Você digitou:

```
Este ser o primeiro cap tulo do meu livro.
Ser um livro mais famoso que o livro de Euclides.
```

Já no seu documento \LaTeX , o principal, você tem:

```
%-----
PREAMBULO
%-----

\begin{document}

Estou escrevendo um livro...
\include{capitulo1.tex}
Tenho certeza que vou chegar l !

\end{document}
```

Ao processar o texto, o resultado obtido, será

Estou escrevendo um livro...
:
Este será o primeiro capítulo do meu livro.
Será um livro mais famoso que o livro de Euclides.
:
Tenho certeza que vou chegar lá!

Onde no lugar das reticências verticais, ocorre uma mudança de página. Vale ressaltar ainda que, caso você salve este arquivo numa pasta diferente do arquivo principal, deve ser especificado o caminho completo do arquivo que será incluído.

6.3 Capa

Aqui daremos apenas uma idéia, para uma capa simples. Com o tempo você será capaz de fazer capas espetaculares, inclusive com figuras. Crie um arquivo com o nome `capa.tex` e salve na mesma pasta do seu arquivo principal.

```
\title{{\Huge Meu primeiro Livro}}
\author{Eu mesmo}
\maketitle
```


O comando `\maketitle` faz com que os demais comandos se arrumem na página. Repare que a data da compilação do documento \LaTeX também aparecerá na capa. Caso você não queira a data na capa, basta usar o comando

```
\date{Outra coisa que n o seja a data}
```

e esta frase aparece em vez da data. No seu documento \LaTeX (o principal), dentro do ambiente `{document}` você digita o comando

```
\include{capa}
```

Se quiser algo mais “com a cara do autor”, você pode simplesmente digitar no arquivo `capa.tex` tudo aquilo que você gostaria de mostrar na capa com as devidas formatações.

6.4 Partes

Suponha que seu livro seja sobre dois grandes assuntos. Então você pode dividi-lo em duas *partes* onde cada parte terá vários capítulos e cada capítulo várias seções. O comando

```
\part{T tulo}
```

faz essa primeira divisão criando uma página que contém apenas o nome da parte.

6.5 Capítulos

Dentro de cada parte (não é necessário que existam as “partes”) podemos incluir vários capítulos. Para isto basta incluir o comando

```
\chapter{Nome do Cap tulo}
```

Automaticamente é gerado o número do capítulo e a formatação do título. Se por algum motivo você não quiser que apareça o número do capítulo, basta digitar um asterisco, logo após a palavra `chapter`.

```
\chapter*{Nome do Cap tulo}
```

O problema é que com isso, o capítulo em questão não aparece no índice. Se mesmo assim, você não quiser o número do capítulo, basta usar o comando:

```
\addcontentsline{toc}{chapter}{nome}
```

com esse comando, o número dessa página aparece no sumário, mesmo sendo um capítulo não numerado.

Agora, como já falamos na questão da organização, basta criar um arquivo, por exemplo, `cap1.tex` e digitar:

```
%---capitulo 1
```

```
Este o primeiro cap tulo do maior livro de todos os tempos.  
O meu Livro!
```

E no principal, teremos

```
%-----
```

```
PREAMBULO
```

```
%-----
```

```
\begin{document}
```

```
\include{capa}
```

```
\part{Unidade I}
```

```
\include{cap1}
```

```
\end{document}
```

Se você já tiver vários capítulos prontos, digamos, uns cinco, divididos em duas partes, o documento \LaTeX principal, fica:

```
%-----
PREAMBULO
%-----

\begin{document}
\include{capa}
\part{Unidade I}
\include{cap1}
\include{cap2}
\include{cap3}
\part{Unidade II}
\include{cap4}
\include{cap5}
\end{document}
```

6.6 Seções

Basicamente funciona como o capítulo, sendo que o seu comando é

```
\section{Nome da Seção}
```

Automaticamente são gerados os números das seções com numeração acompanhando as do capítulo e a formatação do título. Para que não apareça o número da seção, também basta digitar um asterisco, logo após a palavra `section`.

```
\section*{Nome da Seção}
```

Isso também faz com que a seção em questão não apareça no índice. Para resolver isso, basta usar:

```
\addcontentsline{toc}{section}{nome}
```

Você pode fazer as seções dentro dos arquivos `capX.tex` ou pode criar arquivos exclusivos para seções e os chamar dos arquivos `capX.tex`. Por exemplo, suponha que no capítulo 2 do seu livro, você tenha as seções

- A história da minha vida até 10 anos;
- A história da minha vida entre 10 e 15 anos;
- A história da minha vida a partir dos 15 anos.

Você pode criar arquivos separados para cada seção, suponhamos, `ate10.tex`, `de10a15.tex` e `depoisde15.tex`. O primeiro arquivo, `ate10.tex` será:

```
%---seção que fala sobre minha vida até os 10 anos
\section{A história da minha vida até os 10 anos}
Só muitas emoções...
```

O segundo arquivo, `de10a15.tex` será:

```
%---seção que fala sobre minha vida entre os 10 anos e 15 anos
\section{A história da minha vida entre 10 e 15 anos}
Eu passava o dia jogando bola e comendo...
Muitas emoções ainda...
```

E o terceiro arquivo, `depoisde15.tex` será:

```
%----se o que fala sobre minha vida entre os 10 anos e 15 anos
\section{A história da minha a partir dos 15 anos}
Agora eu tenho muitas contas para pagar,
Tenho que trabalhar...
Menos emoções, mais confusões...
```

O arquivo cap2.tex será

```
%---Capítulo 2, a História da minha vida.
\chapter{A história da minha vida}
\input{ate10}
\input{de10a15}
\input{depoisde15}
```

Como resultado, teremos:

Capítulo 2

A história da minha vida

2.1 A história da minha vida até os 10 anos

São muitas emoções...

2.2 A história da minha vida entre 10 e 15 anos

Eu passava o dia jogando bola e comendo... Muitas emoções ainda...

2.3 A história da minha a partir dos 15 anos

Agora eu tenho muitas contas para pagar, Tenho que trabalhar... Menos emoções, mais confusões...

6.6.1 Subseções

Podemos ainda, seguindo o mesmo raciocínio, definir subseções e subsubseções através dos comandos:

- `\subsection;`
- `\subsubsection.`

6.7 Sumário

Quando estamos na digitação de uma monografia ou livro, talvez a maior “dor de cabeça” seja a construção do índice. Alguns editores de texto (visuais) já trazem opções para construir índices e também atualizações rápidas. A questão é que deve-se ir ao texto e “ajeitar no braço” o formato das frases que vão aparecer no texto.

No texto em \LaTeX já vimos que um comando resolve o problema de se determinar os itens que aparecerão no índice:

- `\part;`
- `\chapter;`
- `\section;`
- `\subsection;`
- `\subsubsection.`

Para que apareça o índice, faz-se necessário digitar no ambiente `document` o comando:

```
\tableofcontents
```

6.8 Lista de Figuras

6.9 Referências Bibliográficas

Na produção de monografias ou livros, obviamente existe a pesquisa a outros autores e a ética e o bom senso mandam que se coloque em alguma parte de seu trabalho, os títulos e autores das obras utilizadas. É o que chamamos de *Referências Bibliográficas*.

Seguindo o princípio da organização, deve-se criar um arquivo exclusivo para as Referências Bibliográficas. Além disso, isso deve ser feito de maneira a facilitar a citação durante toda monografia ou livro que se está produzindo. Chamemos esse arquivo de

```
bib.tex
```

Nele teremos:

```
\addcontentsline{toc}{chapter}{Referências Bibliográficas}
\begin{thebibliography}{n}

\bibitem[1]{autor_1} cita o_1.

\bibitem[2]{autor_2} cita o_2.

\bibitem[3]{autor_3} cita o_3.
...

\bibitem[n]{autor_n} cita o_n

\end{thebibliography}
```

O parâmetro `n` junto ao `\begin{thebibliography}` define a quantidade de citações que se irá fazer no trabalho. Em cada item (`\bibitem`) há dois campos:

```
\bibitem[marca]{autor_x}
```

o campo `marca` indica o que será impresso no texto a cada vez que for feita referência ao autor `autor_x`. Por exemplo, se um dos itens for:

```
\bibitem[EL]{elon} LIMA, Elon Lages. Álgebra Linear. Rio de Janeiro, 1998.
```

Toda vez que quisermos citar a obra acima, digitamos

```
(...) de acordo com \cite{elon}, temos (...)
```

o que produz

```
(...) de acordo com [EL], temos (...)
```

OBSERVAÇÃO: O arquivo `bib.tex` deve ser o último arquivo da lista no arquivo principal.

6.10 Índice Remissivo

Em alguns livros, as últimas páginas trazem o número da página em que algumas palavras importantes para o contexto da obra aparecem, é o *Índice Remissivo*. Por exemplo, suponha que seu livro fale sobre determinantes. Provavelmente esta palavra vai aparecer várias vezes no texto e talvez seja interessante saber em quais páginas ela aparece.

Para que uma palavra apareça no índice remissivo, deve-se escrever imediatamente após a ela o comando

```
\index{palavra}
```

Por exemplo, suponha que se queira a palavra *Corinthians* no índice remissivo desse curso de introdução ao \LaTeX . Então, esta frase no arquivo fonte, deve estar da seguinte maneira:

```
Por exemplo, suponha que se queira a palavra Corinthians\index{Corinthians}
no índice remissivo desse curso de introdução ao \LaTeX. Então, esta frase no
arquivo fonte, deve estar da seguinte maneira:
```

Se quiséssemos falar sobre o goleiro do Corinthians, então escreveríamos o comando

```
\index{Corinthians!goleiro}
```

logo após a palavra **goleiro**, para indicar uma subcategoria da palavra *Corinthians*.

Para que o Índice Remissivo apareça no final do trabalho, basta inserir o pacote

```
\usepackage{makeidx}
```

e o comando

```
\makeindex
```

ambos no pré-âmbulo e o comando `\printindex` depois do comando que insere as Referências Bibliográficas.

Referências Bibliográficas

- [1] OETIKER, Tobias, et. al. *Uma não tão pequena introdução ao \LaTeX* . Tradução portuguesa por Alberto Simões. 2007.
- [2] SANTOS, Reginaldo. *Introdução ao \LaTeX* . Departamento de Matemática, UFMG. 2008.
- [3] ----- . *Apostila de \LaTeX* . Programa Especial de Treinamento. Telecomunicações UFF. 2004.
- [4] ANDRADE, Lenimar Nunes. *Breve Introdução ao \LaTeX* . UFPB. 2001.
- [5] KNUTH, Donald E. *The \TeX Book*. Stanford University, 1993.

Índice Remissivo

Índice Remissivo, 53

bibitem, 52

Capítulos, 49

Capa, 48

Corinthians, 53
goleiro, 53

Editores Lógicos, 11

Editores Visuais, 11

enumerate, 34

EPS, 41

Fórmula

Centralizada, 23

junto ao texto, 23

numerada, 23

Figura

ângulo de rotação, 41

altura da, 41

largura da, 41

legenda, 41

proporção da, 41

GnuPlot, 43

Itálico, 17

Kile, 12

Layout da página, 13

Lista de Tabelas, 40

Negrito, 17

packages, 12, 14

pacotes, 14

Partes, 49

preâmbulo, 12

Referências Bibliográficas, 52

Seções, 50

subseções, 51

subsubseções, 51

Subescrito, 17

Sumário, 52

Tamanho da Letra, 13, 17

Texmaker, 12

Tipo de Papel, 13

XFig, 43