
APOSTILA DE L^AT_EX

Programa de Educação Tutorial
Engenharia de Telecomunicações

PETele))

Universidade Federal Fluminense

Niterói-RJ

Maio / 2008

Prefácio

Tendo em vista as diretrizes do MEC em Pesquisa, Ensino e Extensão, o Programa de Educação Tutorial do curso de Engenharia de Telecomunicações da Universidade Federal Fluminense (UFF) desenvolveu um projeto de elaboração de apostilas, com o intuito de auxiliar os alunos do curso no aprendizado de temas importantes a sua formação, mas ausentes em quaisquer ementas de disciplinas; e, além disso, servir de material didático para os cursos de capacitação que são dados pelos alunos do programa para os corpos docente e discente da graduação.

Abaixo segue a lista de apostilas preparadas neste projeto:

HTML Linguagem de programação para hipertextos, principalmente empregada na construção de páginas da Internet (webpages).

LaTeX Sistema de edição de texto largamente utilizado em meios acadêmicos e científicos, bem como por algumas editoras nacionais e internacionais.

LINUX Introdução ao sistema operacional LINUX.

Linguagem C : Linguagem de programação amplamente utilizada em problemas de engenharia e computação.

MATLAB Ambiente de simulação matemática, utilizado em diversas áreas profissionais.

SPICE Ambiente de simulação de circuitos elétricos (analógicos e digitais), utilizado em projeto de circuitos discretos e integrados.

Esta apostila destina-se a introduzir o usuário ao editor de texto LaTeX, explicando como elaborar um documento com os comandos básicos do LaTeX, e não como instalar o programa LaTeX.

Os comandos tratados nesta apostila permitem ao usuário elaborar um bom documento, porém esta apostila não pretende abordar todos comandos existentes devido as diferentes áreas onde o LaTeX pode ser usado e a quantidade de funções que são criados a todo momento (veja Seção 2.1.1). Para maiores informações e um estudo mais aprofundado ao LaTeX consulte as referências bibliográficas no fim do documento.

Autor atual: Thiago Muniz de Souza

Últimas atualizações: Rodolfo Almeida Reis Quarto
José Luiz Gomes Ramos

Este documento é de distribuição gratuita, sendo proibida a venda de parte ou da íntegra do documento.

Sumário

Prefácio	i
1 O que é o \LaTeX ?	3
2 Documento	5
2.1 Estrutura	5
2.1.1 Pacotes	6
2.2 Texto	7
2.2.1 Acentuação	7
2.2.2 Sentenças e Parágrafos	8
2.2.3 Alinhamento	8
2.2.4 Símbolos especiais	9
2.2.5 Traços (-)	9
2.2.6 Estilos de letras e Tamanhos	9
2.2.7 Prevenindo a quebra de palavras	10
2.2.8 Notas	10
2.2.9 Fórmulas	10
2.2.10 Comentário	11
2.2.11 Título do documento	11
2.2.12 Resumo	11
2.2.13 Seções	11
2.2.14 Identação	12
2.2.15 Listas	12
2.2.16 Versos	13
2.2.17 Símbolos	14
2.2.18 Textos pré-formatados	14
3 O ambiente matemático	15
3.1 Onde fica a fórmula ?	15
3.2 Construindo fórmulas	16
3.2.1 Subscritos e Sobrescritos	16
3.2.2 Frações	16
3.2.3 Raízes	16
3.2.4 Símbolos matemáticos	16
3.2.5 Funções	16
3.2.6 <i>Array</i>	17
3.2.7 Delimitadores	18

3.2.8	Fórmulas em várias linhas	19
3.2.9	Linhas	20
3.2.10	Empilhando	21
3.2.11	O comando <i>phantom</i>	21
3.2.12	Espaçamento nas fórmulas	22
3.2.13	Teoremas	22
3.2.14	Tipos especiais de letras	23
4	Tabelas	24
4.1	<i>Tabbing</i>	24
4.2	<i>Tabular</i>	25
4.2.1	Tabelas Longas	27
5	Movendo informações	29
5.1	Referência cruzada	29
5.2	Dividindo o arquivo	30
5.3	Bibliografia	30
5.4	Figuras	33
5.4.1	Subfiguras	35
5.5	Sumário	36
5.6	Índice	36
5.7	Tipos de arquivos	38
6	Estrutura visual	39
6.1	Cabeçalho e Rodapé da página	39
6.2	Área de impressão	42
6.3	Espaços e Medidas	42
6.4	Caixas	45
6.5	Cores	45
6.6	Minipage	46
6.7	Novos comandos e ambientes	47
A	Utilizando o LaTeX através de um Terminal de Comando	49
B	Símbolos matemáticos	51
C	Outros símbolos	54
	Referências Bibliográficas	55

Capítulo 1

O que é o \LaTeX ?

Antes de saber o que é o \LaTeX é preciso conhecer o \TeX . O \TeX é um programa criado por Donald Knuth, na década de 70, com a finalidade de aumentar a qualidade de impressão com base nas impressoras da época. Ele é utilizado para processar textos e fórmulas matemáticas.

\LaTeX é um programa que reúne comandos que utilizam o \TeX como base de processamento. Foi criado por Leslie Lamport na década de 80 com o objetivo de facilitar o uso do \TeX através de comandos para diferentes funções.

É um editor de textos especialmente voltado para a área matemática, contendo comandos para montar as mais diversas fórmulas. Gera textos de alta qualidade tipográfica (espaçamento entre palavras, combinação de letras etc.), além de ser muito bom para fazer textos grandes como livros.

O \LaTeX é um processador baseado no estilo lógico. Os programas de processamento de texto podem ser divididos em duas categorias:

Estilo visual Nestes processadores de texto, existe um menu na tela apresentando os recursos que podem ser usados, sendo selecionados através do mouse. O texto que você digita aparece na tela da mesma forma que vai ser impresso. Isso é conhecido como WHAT-YOU-SEE-IS-WHAT-YOU-GET (WYSIWYG). Ex: Microsoft Word®.

Estilo lógico Nesta categoria, o processamento é feito em duas etapas distintas:

- O texto a ser impresso e os comandos de formatação são escritos em um arquivo fonte com o uso de um editor (isso não impede que também haja um menu na tela onde os comandos podem ser selecionados, isto é, apenas um adicional oferecido por fabricantes para facilitar a digitação).
- Em seguida este arquivo é compilado e gera um arquivo de saída que pode ser visualizado. Ex: HTML, DVI, PDF etc.

Os comandos \LaTeX foram criados com base em diversos estudos sobre diagramação. Isto foi importante para fazer com que o \LaTeX entenda o que o autor deseja fazer, por exemplo, diferenciar um texto matemático de uma citação de fala. Segundo os estudos, existem formas que permitem tornar o texto muito mais claro. O tamanho deve ter um limite ideal para facilitar a leitura, assim como a fonte. O espaçamento das linhas, letras e palavras também tem uma medida ideal.

Geralmente, quando o autor está trabalhando com um processador visual, ele comete muitos erros por não conseguir conciliar uma boa estética com uma estrutura lógica bem compreensível. Utilizando o \LaTeX , que é um processador lógico, o autor se preocupará mais com

o conteúdo. Dessa forma, seu texto não terá apenas uma boa estética e estrutura coerente, mas também um bom conteúdo. Além disso, com a troca de apenas um comando, algo pode ser mudado futuramente com facilidade, o que deixa o documento muito mais flexível.

Vamos ver o que acontece tecnicamente:

O autor escreve seu documento usando os comandos do \LaTeX . O \LaTeX entende o que o autor quis dizer e transforma os comandos digitados em uma linguagem inteligível pelo \TeX , ou seja, o texto escrito em linguagem \LaTeX é processado por um compilador seguindo as regras dessa linguagem, isto é, transforma um arquivo *.tex (que possui o código *tex*) em um arquivo *.dvi (*device independent*). Este último é o documento pronto para ser visualizado.

Claro que não se pode esquecer que seu computador deve ter a biblioteca do \LaTeX para fazer isso. Alguns dos sistemas \TeX , atuais, disponíveis no mercado são: fptex, pctex, miktex (Windows) e tetex (Linux, embora provavelmente já o tenha instalado).

Esse arquivo *.dvi pode ser lido independentemente da versão do editor usado, logo pode ser lido em qualquer sistema, contanto que ele tenha um programa específico para lê-lo.

Outra vantagem do \LaTeX é sua estabilidade, ou seja, a probabilidade de se encontrar um *bug* nele é mínima e justamente por ser *free software* seu sistema é aberto, o que permite que qualquer um corrija possíveis *bugs* ou que possa adaptá-lo às suas necessidades. A cada momento surgem novos pacotes com funções criadas por usuários espalhados por todo o mundo.

Capítulo 2

Documento

2.1 Estrutura

A linguagem LaTeX funciona à base de comandos que são iniciados com `\`, que é um marcador (*tags*, de *Tag languages*).

Os comandos são escritos nas formas `\comando` ou `\begin{comando}... \end{comando}`. Quando vem escrito nesta última forma, ele é chamado de **ambiente**.

O texto de cada tipo de documento começa com `\begin{document}` e termina com `\end{document}`. Tudo o que vem antes disso é considerado o **preâmbulo** e tudo o que vem depois de `\end{document}` é ignorado.

É no preâmbulo que são colocadas todas as informações referentes às principais características que terá seu documento. Começa com `\documentstyle{estilo}` no caso do \LaTeX 2.09 e com `\documentclass{estilo}` no caso do \LaTeX 2_ε (segunda edição).

No lugar de estilo é colocado o nome de um dos estilos pré-definidos, como:

article Textos pequenos;

report Relatórios;

book Livros, apostilas;

letter Cartas.

Obs: Os estilos não são apenas estes. Geralmente congressos, universidades e outros meios disponibilizam outros estilos de formatação para apresentação de trabalhos. Isso mostra uma das vantagens do LaTeX, que é a flexibilidade para se criar novas formatações que atendam à diferentes necessidades.

Podem, também, ser selecionadas algumas opções dentro do estilo escolhido, como:

- **Tamanho**: Padrão da letra: 11pt ou 12pt(pontos), o último é usado com mais frequência;
- **twoside**: Imprime em ambos os lados da página;
- **oneside**: Imprime em um só lado da página;
- **twocolumn**: Produz o texto disposto em duas colunas na página;

- **onecolumn**: Produz o texto disposto em uma coluna;
- **landscape**: Produz uma página na forma de paisagem;
- **leqno**: Faz com que a numeração das fórmulas seja colocada à esquerda em vez de à direita;
- **fleqn**: Faz com que as fórmulas fiquem localizadas na margem esquerda em vez de estarem centralizadas;
- **openright**: Faz com que os capítulos sejam iniciados apenas nas páginas ímpares;
- **openany**: Permite que os capítulos sejam iniciados nas páginas ímpares ou pares.
- **Tamanho da folha**: Pode ser a4, letterpaper etc..

Essas opções são colocadas entre colchetes, sem espaço entre as palavras e com vírgula.
Ex:

```
\documentstyle[twocolumn,12pt,a4]{article}
\usepackage{pacote}
\begin{document}
.
.
.
\end{document}
```

Obs: caso a opção *twocolumn* não tenha sido declarada no preâmbulo, existe o comando `\twocolumn` que produz o texto em duas colunas a partir do ponto onde foi colocado, iniciando uma nova página. Para reverter ao modo inicial, utilize o comando oposto; no caso, o `\onecolumn`.

2.1.1 Pacotes

Pode-se definir pacotes como um conjunto de arquivos que implementam uma determinada característica adicional para os documentos escritos em LaTeX.

Quando o usuário quiser montar um documento um pouco mais elaborado, perceberá que os comandos básicos que o LaTeX contém não são suficientes, sendo necessário o uso de algo que aumente a sua capacidade.

Alguns pacotes já vêm como distribuição básica do LaTeX, outros podem ser encontrados separadamente (veja a referência [4]) pois a todo momento novos pacotes são criados por usuários em todo o mundo.

Estes pacotes são inseridos no preâmbulo usando o comando

```
\usepackage[opções]{pacote}
```

Ao longo de toda a apostila serão apresentados pacotes com diferentes funções. Abaixo, segue a lista com uma breve descrição de alguns deles. O funcionamento de cada um será explicado posteriormente.

Os principais são:

graphicx Para inserir gráficos. Veja seção 5.4;

color Para usar cor no seu texto. Veja seção 6.5;

babel Para traduzir nomes que aparecem em inglês na estrutura do documento. Ex: *chapter*, *section*, *tableofcontents*, etc. Neste caso, para que estas palavras sejam traduzidas para o português brasileiro, use a opção [brazil].

fontenc Permite que o LaTeX compreenda a acentuação feita direto pelo teclado. É usado com o opcional [T1].

amsfonts Define alguns estilos de letras para o ambiente matemático;

fancyhdr Para fazer cabeçalhos personalizados. Veja seção 6.1.

Obs: Nem todos os pacotes são compatíveis com qualquer versão do LaTeX. Os criadores, sempre que criam novos pacotes, tentam deixá-los compatíveis com qualquer versão do LaTeX, porém algumas vezes isso não é possível.

É muito simples saber se seu sistema possui determinados pacotes, ou instalá-los: basta consultar o manual dele. Caso o usuário esteja escrevendo algum documento e precise mudar constantemente de computador, mas não sabe se o sistema deste possui o pacote que seu documento precisa, basta copiar todos os arquivos do pacote e deixar no mesmo diretório em que seu documento está. Quando o LaTeX está compilando o documento, o arquivo do pacote será procurado no caminho padrão do sistema ou no próprio diretório do seu documento.

Por exemplo, imagine que você esteja escrevendo um trabalho e que ele deva seguir um determinado modelo que um congresso exija. Provavelmente, seu sistema não terá instalado o pacote que faz isto. O que fazer então? É só pegar os arquivos deste pacote e deixar no mesmo diretório do arquivo do trabalho. Quando compilar, o LaTeX lerá os arquivos deste pacote e gerará o documento no formato desejado. E se for preciso mudar o sistema operacional, não haverá problema algum, basta andar sempre com estes arquivos (que são pequenos, poucos *kilobytes*) junto com os arquivos de seus documentos.

Cada pacote possui um manual com os comandos e suas funções. Alguns dos pacotes descritos nesta apostila foram explicados de acordo com o manual deles, disponíveis no próprio sistema. Lembrando, basta um pouco de curiosidade para descobrir a quantidade de recursos oferecidos pelo LaTeX. Basta ler o manual do sistema e descobrir como trabalhar com pacotes. Há uma infinidade deles com as mais diferentes funções.

2.2 Texto

2.2.1 Acentuação

Quando o pacote *fontenc* não tiver sido declarado, a acentuação no LaTeX é feita com comandos da seguinte maneira:

- $\backslash\{c\}$ – ç
- $\backslash\{e\}$ – è
- $\backslash\prime\{e\}$ – é
- $\backslash\wedge\{e\}$ – ê

- `\~{e}` – ã
- `\"q` – q̈

Outros tipos de acentos estão no Apêndice C.

2.2.2 Sentenças e Parágrafos

Estamos acostumados a visualizar o espaçamento entre palavras de acordo com o número de vezes em que apertamos a tecla de espaçamento. Já no LaTeX, isso não importa, dado que sempre será contado apenas um, independentemente da quantidade de espaços inseridos.

O primeiro parágrafo será iniciado sem indentação, como pode ser observado logo acima. O comando `\indent` adiciona uma largura igual ao tamanho da indentação do parágrafo normal e o comando `\noindent` retira a indentação do local onde ela deveria aparecer. Estes comandos funcionam somente para alguns estilos.

Para criar um novo parágrafo, basta pular uma linha ou utilizar o comando `\par` no lugar em que será iniciado o novo parágrafo.

Para passar para a linha abaixo da qual se está digitando, coloque `\\` e a linha será quebrada neste ponto. Ex:

<code>observe que esta linha está\\</code>	<code>observe que esta linha está</code>
<code>quebrada após o\\</code>	<code>quebrada após o</code>
<code>uso das duas barras.</code>	<code>uso das duas barras.</code>

Usando o comando `\linebreak` a linha é quebrada e a parte anterior ao comando fica justificada.

Ex:

Esta linha está `\linebreak` justificada.

Esta	linha	está
justificada.		

Quando o comando `\newpage` é usado, o texto passa para a próxima página. O comando `\pagebreak[num]` força a quebra da página, onde o argumento opcional `[num]` é um valor inteiro de 1 a 4 que define a prioridade do funcionamento do comando, onde 4 é o maior valor. O comando `\nopagebreak[num]` faz o inverso de `pagebreak`, ou seja, impede que a página seja quebrada no local onde o comando foi colocado.

Obs: Os comandos `*` e `\\[medida]` tem a mesma função do comando `\\`. A diferença está no fato de que `*` impede que o texto mude de página na quebra daquela linha e que `\\[medida]` quebra a linha, porém acrescentando o espaço que está determinado entre colchetes. Veja as unidades de medida na Seção 6.3.

2.2.3 Alinhamento

Usando o ambiente `center` o texto poderá ser centralizado.

Com o ambiente `flushleft` o texto é alinhado a esquerda.

E com o ambiente `flushright` o texto é alinhado a direita.

<code>\begin{center}</code> texto <code>\end{center}</code>	texto
<code>\begin{flushleft}</code> texto <code>\end{flushleft}</code>	texto
<code>\begin{flushright}</code> texto <code>\end{flushright}</code>	texto

Obs: As linhas são apenas para facilitar a visualização.

2.2.4 Símbolos especiais

O LaTeX possui 10 caracteres especiais com os quais são digitados comandos:

~ ^ \ # \$ % & _ { }

Para que eles apareçam no texto, coloque \ na frente. Veja a tabela de símbolos no apêndice.

2.2.5 Traços (-)

Podem ser feitos três diferentes tamanhos de traços se digitados uma, duas ou três vezes (-)

Ex:

- com -
 – com --
 — com ---

2.2.6 Estilos de letras e Tamanhos

As palavras podem ser colocadas em:

<code>\textbf{negrito}</code>	negrito
<code>\textsf{sans serif}</code>	sans serif
<code>\textsl{slanted}</code>	<i>slanted</i>
<code>\textsc{small caps}</code>	SMALL CAPS
<code>\texttt{letra de máquina}</code>	letra de máquina
<code>\textrm{romano}</code>	romano

Os comandos `\bf`, `\it`, `\sf`, etc..., também podem ser usados e fazem parte do L^AT_EX 2.09.

O texto inteiro também pode ter o tipo de letra diferente do romano, que é o padrão. Basta usar os comandos:

- `\sffamily` - Para colocar o texto no tipo sans serif
- `\ttfamily` - Para colocar o texto no tipo letra de máquina

- `\rmfamily` - Para colocar o texto no tipo romano

A partir do ponto onde estes comandos forem colocados, será mudado o estilo da letra.

Obs: É possível que seu sistema possua outras fontes, basta consultar os pacotes que estão instalados em seu computador.

E os tamanhos podem ser:

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

Usando os comandos para fazer ambientes pode-se mudar o tamanho e a forma das palavras em vários parágrafos. Ex: `\begin{huge}... \end{huge}`.

2.2.7 Prevenindo a quebra de palavras

Pode acontecer quebra de alguma palavra na troca de linha ou página de forma errada, pois o LaTeX não utiliza a regra de hifenização das palavras em português. Para evitar isso use `\mbox{palavra}`.

Há também outra maneira: colocar no preâmbulo o comando `\hyphenation{pa-la-vra}` com a palavra dividida em sílabas da forma correta.

Mas é recomendável que só se faça este tipo de coisa quando tiver sido terminado o texto, pois conforme o texto vai sendo digitado a disposição deste na tela pode mudar.

2.2.8 Notas

As notas de rodapé de página são produzidas com o comando `\footnote{texto}`

Ex: Esta frase tem uma nota no fim da página ¹.

Há também uma maneira de colocar as notas na margem da página. Basta colocar o comando

`\marginpar{texto}`. A nota ficará na altura da linha de texto em que foi colocada.

Ex: Esta sentença tem uma nota `\marginpar{esta ... página}` na margem.

2.2.9 Fórmulas

Nos textos, podem ser inseridas fórmulas com os seguintes comandos:

- `\(fórmula\)`

¹esta é a nota do pé da página

esta
é
a
nota
na
mar-
gem
da
pá-
gina

- $fórmula$
- `\begin{math}fórmula\end{math}`

A maneira como as fórmulas são feitas e o resultado do uso de cada comando serão vistos no capítulo 3.

2.2.10 Comentário

Utilizando o caracter % no início de uma linha do código fonte de documento o LaTeX ignora o que está escrito nela na hora de compilar. Esse caracter é considerado um marcador de comentário.

Ex:

```
% este é o comentário no código fonte
```

2.2.11 Título do documento

É feito com os seguintes comandos:

```
\title{título}
\author{autor}
\date{data} -> este é opcional, se não usar é
                colocada a data atual
\begin{document}
\maketitle
```

Caso haja mais de um autor, pode ser colocado da seguinte forma:

```
\author{primeiro \and segundo}
```

Outra maneira de se fazer o título é usando o ambiente *titlepage* que é colocado após o ambiente *document*. Neste ambiente há a liberdade de montar a capa da maneira desejada, sem precisar usar os comandos citados acima.

2.2.12 Resumo

Um parágrafo com o título de resumo pode ser feito usando o ambiente *abstract*. Geralmente, ele é colocado na página de título ou em uma página separada para que o autor dê uma breve explicação sobre o documento. Ele só é válido nos estilos *report* e *article*.

2.2.13 Seções

Em textos um pouco mais longos pode haver várias seções. Então, o LaTeX contém alguns comandos para dividir seu texto, deixando-o mais organizado e com estrutura coerente. São eles:

```

\part{parte}
\chapter{capítulo}
\section{seção}
\subsection{sub-seção}
\subsubsection{sub-sub-seção}
\paragraph{parágrafo}
\subparagraph{subparágrafo}

```

Todas essas partes e sub-partes são numeradas seguindo uma estrutura lógica. Mas colocando um asterisco após o comando, não ocorre a numeração.

Ex: `\section*{seção}`.

Obs: Os comandos *part* e *chapter* só podem ser usados com os estilos **report** ou **book**.

2.2.14 Identação

Para facilitar a leitura de algumas sentenças, é necessário que as enfatize, o que é feito através do ambiente:

```

\begin{quote}
    sentença que está enfatizada
\end{quote}

```

Ex: Esta é uma das célebres frases de um dos maiores cientistas do século passado: Albert Einstein.

"O importante é não parar de questionar. A curiosidade tem sua própria razão para existir. Uma pessoa não pode deixar de se sentir reverente ao contemplar os mistérios da eternidade, da vida, da maravilhosa estrutura da realidade. Basta que a pessoa tente apenas compreender um pouco mais desse mistério a cada dia. Nunca perca uma sagrada curiosidade".

Também pode ser usado o comando:

```

\begin{quotation}
    sentença que está enfatizada
\end{quotation}

```

A diferença do anterior deste é que o último permite que sejam enfatizados vários parágrafos devido a sua indentação.

2.2.15 Listas

Na hora de se construir itens, pode-se perceber uma das grandes facilidades proporcionadas pelo LaTeX, pois há comandos próprios para isso.

Para listas:

- Somente com marcação, usa-se:

```

\begin{itemize}
  \item primeiro item           – primeiro item
  \item segundo item           – segundo item
  \item terceiro item          – terceiro item
\end{itemize}

```

É permitido colocar alguns símbolos para fazer um marcador personalizado no item, como: ♡, ◇, ♠, ‡, §, †. Eles são colocados assim: `\item[comando do símbolo] texto`. Veja como fazer estes símbolos no Apêndice B.

```

\begin{itemize}
  \item[♣\clubsuit$]item com marca-
    dor personalizado.
\end{itemize}

```

- Listas numeradas usa-se:

```

\begin{enumerate}
  \item primeiro item           1. primeiro item
  \item segundo item           2. segundo item
  \item terceiro item          3. terceiro item
\end{enumerate}

```

- Para listas com descrição é usado:

```

\begin{description}
  \item[item] descrição deste   item descrição deste
  \item[item] descrição deste   item descrição deste
\end{description}

```

2.2.16 Versos

Para fazer a construção de um verso basta usar o ambiente `verse`:

```

\begin{verse}
  verso
\end{verse}

```

Ex: Esta é a parte de uma poesia de Gonçalves Dias.

```

\begin{verse}\index{verse}
... \\\
Nosso céu tem mais estrelas\\
Nossas várzeas têm mais flores\\
Nossos bosques têm mais vida\\
Nossa vida mais amores\\
...
\end{verse}

```

...
Nosso céu tem mais es-
trelas
Nossas várzeas têm mais
flores
Nossos bosques têm mais
vida
Nossa vida mais amores
...

Note que deve ser usado `\\` para pular linhas. Note também a forma como acontece a quebra da frase que não cabe na mesma linha.

2.2.17 Símbolos

Alguns símbolos e caracteres de língua estrangeira podem ser gerados com o LaTeX.

Ex:

`\c{o}` = φ

`\S` = \S

`\copyright` = ©

Veja mais símbolos nos apêndices.

2.2.18 Textos pré-formatados

O LaTeX também permite que seja digitado algo da mesma forma que deverá aparecer na tela ou trechos de texto que possuem muitos caracteres. Isso é feito com o ambiente *verbatim*. Dentro desse ambiente pode ser digitado qualquer coisa, até mesmo os espaços são colocados da mesma forma. Isto é muito útil na hora de digitar textos na forma de uma linguagem de programação, por possuir muitos caracteres que também são usados para fazer comando em LaTeX.

Usando o ambiente, seu texto ficará evidenciado em uma linha a parte, mas para que ele continue na mesma linha em que está sendo digitado, use o comando `\verb= seu texto =`, onde este sinal de igual pode ser substituído por qualquer coisa desde de que não seja espaço, asterisco ou letras.

Ex: texto `\verb+{|@#$$+ texto`, você verá: texto `{|@#$$ texto`.

Se for usado *verbatim* ou `\verb` seguido de um asterisco, em vez dos espaços em branco será colocado o símbolo: `_`.

Ex: `\verb*=a b c d= a_b_c_d`

É importante lembrar que dentro do ambiente *verbatim* o comando que você colocar não será considerado.

Capítulo 3

O ambiente matemático

3.1 Onde fica a fórmula ?

As fórmulas matemáticas podem ser digitadas tanto no meio de um texto quanto em destaque:

- No meio do texto:

Segundo a equação:
`$a^{2}= b^{2}+c^{2}$`
concluimos que...

Segundo a equação: $a^2 = b^2 + c^2$ concluimos que...

Deve ser usado `$...$` para que a equação apareça no meio do texto. Além disso, podem ser usados:

`\(fórmula \)` ou `\begin{math} fórmula \end{math}`

- Em destaque:

Segundo a equação:
`\begin{equation}`
`a^{2}= b^{2}+c^{2}`
`\end{equation}`
podemos concluir que...

Segundo a equação:
$$a^2 = b^2 + c^2 \quad (3.1)$$

podemos concluir que...

Outra maneira para fazer a equação aparecer em destaque é usando os ambientes:

`\[fórmula \]` ou `\begin{displaymath} fórmula \end{displaymath}`

Neste modo a equação é numerada automaticamente de acordo com a seção. Para que isso não aconteça use o comando `\nonumber` dentro do ambiente.

3.2 Construindo fórmulas

3.2.1 Subscritos e Sobrescritos

Sobrescrito – É feito usando: $b^{\{e\}}$ onde b é a base e e o expoente.

Ex: $2^{\{5\}} \rightarrow 2^5$

Subscritos – É feito usando: $b_{\{i\}}$ onde b é a base e i o índice.

Ex: $2_{\{5\}} \rightarrow 2_5$

3.2.2 Frações

Podem ser feitas usando:

- /
Ex: $(a+b)/2 \rightarrow (a + b)/2$
- $\frac{\text{numerador}}{\text{denominador}}$
Ex: $\frac{a+b}{2} \rightarrow \frac{a+b}{2}$

3.2.3 Raízes

São feitas usando: $\sqrt[\]{\ } Ex: \sqrt[3]{8} \rightarrow \sqrt[3]{8}$

Se for omitido o termo [], automaticamente a raiz será quadrada.

3.2.4 Símbolos matemáticos

O LaTeX possui vários símbolos para montar fórmulas como integrais, somatórios, letras especiais etc.

Ex:

$\int \rightarrow \int$

$\exists \rightarrow \exists$

$\infty \rightarrow \infty$

Veja mais no Apêndice B.

3.2.5 Funções

O LaTeX também possui símbolos de funções.

Ex:

$\log 10 \rightarrow \log 10$

$\sin 60 \rightarrow \sin 60$

Veja como usar subscritos em algumas funções como limite e somatório:

```
\[
\lim_{x\rightarrow\infty} \frac{x^3}{e^x}
\]
```

$$\lim_{x \rightarrow \infty} \frac{x^3}{e^x}$$

```
\[
\sum_{n=0}^{\infty} a_n^2
\]
```

$$\sum_{n=0}^{\infty} a_n^2$$

```
\sum_{n=0}^{\infty} a_n^2
```

$$\sum_{n=0}^{\infty} a_n^2$$

Os intervalos são colocados da mesma forma que se põe sobre e subscritos. E repare na diferença da disposição dos intervalos quando é usado fórmula em destaque e no meio do texto.

Ex:

```
\[
f(t)= \frac{A}{2} + \frac{jA}{2 \pi }
\sum_{\stackrel{-}{\infty}}{n
\neq 0} \frac{1}{n} \ , e^{jn2\pi t}
\]
```

$$f(t) = \frac{A}{2} + \frac{jA}{2\pi} \sum_{\substack{-\infty \\ n \neq 0}}^{\infty} \frac{1}{n} e^{jn2\pi t}$$

O comando `\stackrel{-}{\infty}` encontrado no exemplo será visto na seção 3.2.10.

Obs.: no Apêndice B há uma lista com as funções.

3.2.6 Array

É um ambiente que separa os itens em linhas e colunas. A posição do item em relação à sua coluna é especificado por uma simples letra (c-centro, r- direita, l-esquerda). As linhas são separadas usando `\\` e as colunas, com o símbolo `&`. Após a última coluna, não deve ser colocado `&`. Também não se deve esquecer de colocar algo para indicar que é um ambiente matemático.

Ex 1:

```
\begin{equation}
\begin{array}{lr}
x & y \\
z & w \\
\end{array}
\end{equation}
```

$$\begin{array}{lr} x & y \\ z & w \end{array} \quad (3.2)$$

Ex 2:

```
\begin{equation}
\int_{-L}^L \operatorname{sen} \frac{m \pi x}{2} \operatorname{sen} \frac{n \pi x}{2} dx =
\left \{
\begin{array}{cc}
0, & m \neq n \\
1, & m = n
\end{array}
\right.
\end{equation}
```

$$\int_{-L}^L \operatorname{sen} \frac{m\pi x}{2} \operatorname{sen} \frac{n\pi x}{2} dx = \begin{cases} 0, & m \neq n \\ 1, & m = n \end{cases} \quad (3.3)$$

Obs.: os comandos `\left` `\right` encontrados no exemplo serão vistos na seção 3.2.7.

Através dos seguintes comandos opcionais:

- t A primeira linha dentro do ambiente `array` se mantém na mesma altura da linha que antecede ao ambiente e posterior à esta.
- b A última linha dentro do ambiente `array` se mantém na mesma altura da linha que antecede ao ambiente e posterior à esta.

Vamos ver um exemplo para que fique mais claro.

Obs.: Deve ser lembrado que estes são comandos opcionais, logo devem ser colocados entre colchetes.

Ex 3:

```
\[
x -
\begin{array}[t]{c}
a \\
b
\end{array}
-
\begin{array}[b]{c}
x \\
y
\end{array}
\]
```

$$x - \begin{array}[t]{c} a \\ b \end{array} - \begin{array}[b]{c} x \\ y \end{array}$$

3.2.7 Delimitadores

São símbolos que limitam a expressão, como parentêses, chaves e colchetes. É usado `\left` para a limitar parte esquerda e `\right` para a parte direita. Fazendo a combinação destes símbolos com o ambiente `array` podem ser construídas as matrizes.

Ex:

```
\[
    \left(
      \begin{array}{c}
        x \\
        y
      \end{array}
    \right)
\]
```

Ex:

```
\begin{equation}
\frac{d}{dt}\left( \begin{array}{c}
u \\
v
\end{array} \right) = \left( \begin{array}{cc}
-1 & -2 \\
0 & -1
\end{array} \right) \left( \begin{array}{c}
u \\
v
\end{array} \right)
\end{equation}
```

$$\frac{d}{dt} \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} -1 & -2 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \quad (3.4)$$

No lugar de (e) no exemplo acima, poderíamos ter colocado: { , } [,] , |. Quando um ponto (.) for usado após *left* ou *right* não aparecerá delimitador, observe o uso disto na equação 3.3

3.2.8 Fórmulas em várias linhas

O ambiente *eqnarray* é uma combinação do ambiente *array* com o ambiente matemático de equação. Assim como no *array*, também são usados & e \\, com a opção de usar *\nonumber* para não numerar as fórmulas.

Ex 1:

```
\begin{eqnarray}
x & = & m + n + p \\
y & = & z + w + u \nonumber \\
x & & p + n
\end{eqnarray}
```

$$x = m + n + p \quad (3.5)$$

$$y = z + w + u$$

$$x = p + n \quad (3.6)$$

Repare no espaço em branco na última linha entre os &'s e o resultado disso. E também no uso de *\nonumber*.

Para que nenhuma equação seja numerada é só usar o ambiente *{eqnarray*}*.

Ex 2:

```
\begin{eqnarray*}
\int_{-L}^L f(x) dx = a_0 \int_{-L}^L dx +
```

```

\sum_{m=1}^{\infty} a_{m}
\int_{-L}^L \cos \frac{m \pi x}{2} \, dx \ \
&+& \sum_{m=1}^{\infty} b_{m}
\int_{-L}^L \sen \frac{m \pi x}{2} \, dx
\sum_{n=0}^{\infty}
\end{eqnarray*}

```

$$\int_{-L}^L f(x) dx = a_0 \int_{-L}^L dx + \sum_{m=1}^{\infty} a_m \int_{-L}^L \cos \frac{m\pi x}{2} dx + \sum_{m=1}^{\infty} b_m \int_{-L}^L \sen \frac{m\pi x}{2} dx$$

Como visto anteriormente, cada equação recebe uma diferente referência. Porém, se o usuário desejar usar a mesma referência para todas as equações, é só utilizar o pacote chamado *subeqnarray*.

No preâmbulo deve ser colocado:

```
\usepackage{subeqnarray}
```

E no lugar do ambiente *eqnarray* use *subeqnarray*.

Ex:

```

\begin{subeqnarray}\label{equ:sub}
\slabel{sub1} a^2&=&b^2 + c^2\\
\slabel{sub2} a &=&b - 5
\end{subeqnarray}

```

Faz:

$$a^2 = b^2 + c^2 \tag{3.7a}$$

$$a = b - 5 \tag{3.7b}$$

A primeira equação possui o número (\ref{sub1}) 3.7a e a segunda (\ref{sub2}) 3.7b. Já se quisermos nos referir ao conjunto de equações é só usar (\ref{equ:sub}) 3.7.

3.2.9 Linhas

Com o comando `\overline{fórmula}` é criada uma linha acima de uma fórmula e com `\underline{fórmula}` uma linha abaixo da fórmula.

Ex:

$$\begin{aligned} \overline{(\overline{A \cdot B}) + (\overline{A + D})} &= \overline{(\overline{A \cdot B}) \cdot (\overline{A + D})} \\ &= \overline{(\overline{A \cdot B}) \cdot (\overline{A + D})} \\ &= \overline{(\overline{A \cdot B \cdot A}) + (\overline{A \cdot B \cdot D})} \\ &= \overline{A \cdot B \cdot D} \end{aligned}$$

```

\begin{eqnarray*}
\overline{\overline{(\overline{A} \cdot B)}} + \overline{(A + \overline{D})}}
&=&
\overline{\overline{(\overline{A} \cdot B)}}
\cdot
\overline{\overline{(A + \overline{D})}} \ \backslash\backslash
&=&
(\overline{A} \cdot B) \cdot (A + \overline{D}) \ \backslash\backslash
&=&
(\overline{A} \cdot B \cdot A)
+
(\overline{A} \cdot B \cdot \overline{D}) \ \backslash\backslash
&=&
\overline{A} \cdot B \cdot \overline{D}
\end{eqnarray*}

```

Também podem ser feitos outros tipos diferentes de linhas usando:

```

\overbrace{xyz} \rightarrow \overbrace{xyz}
\underbrace{xyz} \rightarrow \underbrace{xyz}
\widetilde{xyz} \rightarrow \widetilde{xyz}
\widehat{xyz} \rightarrow \widehat{xyz}
\overleftarrow{xyz} \rightarrow \overleftarrow{xyz}
\overrightarrow{xyz} \rightarrow \overrightarrow{xyz}
\overline{xyz} \rightarrow \overline{xyz}

```

Há também a possibilidade de se colocar sobre e subscritos fazendo:

```

\overbrace{xyz}^a \rightarrow \overbrace{xyz}^a

```

O comando `\underline` serve para sublinhar e também pode ser usado sem a necessidade de estar em um ambiente matemático.

Ex: `\underline{palavra}` palavra

3.2.10 Empilhando

Usando o comando `\stackrel` é possível colocar um símbolo ou texto acima de outro.

`\stackrel{símbolo acima}{símbolo abaixo}`.

Ex: ${}_{92}^{239}U^* \xrightarrow{24min} {}_{93}^{293}Np^* + {}_{-1}^0\beta$

`$_{92}^{239}U^{*}$ \stackrel{24 min}{\longrightarrow} _{93}^{293}Np^{*} + _{-1}^{0}\beta $`

3.2.11 O comando *phantom*

Este comando é bastante útil quando for desejável escrever algo alinhado em sub e sobrescritos.

Ex:

`U_{ij}^{n} \longrightarrow U_{ij}^n`

$$U_{\{ij\}^n} \longrightarrow U_{ij}^n$$

Note o alinhamento da letra n com as letras i e j , quando é usado o comando *phantom*.

3.2.12 Espaçamento nas fórmulas

No modo matemático o $\text{T}_{\text{E}}\text{X}$ ignora os espaços dados colocando o espaço que convém a ele, mas como alguns autores gostam de mudar isso, há alguns comandos especiais de espaçamento:

$\backslash,$ pequeno espaço $\backslash;$ grande espaço
 $\backslash:$ médio espaço $\backslash!$ espaço negativo(backspace)

É bom deixar o $\text{T}_{\text{E}}\text{X}$ colocar o espaço que ele quer, mas como nem tudo é perfeito, deve-se ficar atento quando houver símbolos de integral, derivada, raízes e quocientes, pois geralmente o $\text{T}_{\text{E}}\text{X}$ confunde a estrutura lógica.

Ex: ydx é visto como o produto de três variáveis pelo $\text{T}_{\text{E}}\text{X}$, logo, quando digitar isso, coloque espaço para que se compreenda que é uma derivada $y dx \rightarrow y \backslash, dx$.

3.2.13 Teoremas

Geralmente um texto matemático possui teoremas, proposições etc. Para isso o $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ tem um comando que define um ambiente com este tipo de estrutura.

Em primeiro lugar deve ser feita uma declaração com o comando $\backslash\text{newtheorem}\{\text{ambiente}\}\{\text{título}\}[\text{numeração}]$, onde *ambiente* é o nome do novo ambiente a ser usado, *título* é uma denominação que irá aparecer como teorema, lei, axioma, etc. e *numeração* é a seqüência da numeração que este ambiente irá receber, como *chapter*, *section*, é opcional. Este comando pode ser colocado em qualquer parte do seu documento.

Em seguida deve ser usado o ambiente com o nome escolhido para escrever o texto.

```
\begin{ambiente}[nome do teorema]
texto
\end{ambiente}
```

Ex:

```
\newtheorem{ambiente}{Lei}
[chapter]
.
.
.
\begin{ambiente}[Lenz]
O sentido da corrente...
origem.
\end{ambiente}
```

Lei 3.1 (Lenz) *O sentido da corrente induzida é aquele que produz um fluxo magnético que se opõe à variação do fluxo magnético que lhe deu origem.*

A numeração seguirá a mesma seqüência quando for usado novamente o mesmo ambiente.

3.2.14 Tipos especiais de letras

Dentro do ambiente matemático também há a possibilidade de se mudar os tipos de letras da seguinte forma:

$\$\mathrm{X Y Z}\$ \rightarrow XYZ$

$\$\mathcal{X Y Z}\$ \rightarrow \mathcal{X}\mathcal{Y}\mathcal{Z}$

$\$\mathbf{X Y Z}\$ \rightarrow \mathbf{XYZ}$

$\$\mathsf{X Y Z}\$ \rightarrow \mathsf{XYZ}$

$\$\mathtt{X Y Z}\$ \rightarrow \mathtt{XYZ}$

$\$\mathit{X Y Z}\$ \rightarrow \mathit{XYZ}$

$\$\mathbb{N I Z Q}\$ \rightarrow \mathbb{NIZQ}$.

Este precisa do pacote *amsfonts*. Para isso basta colocar o comando `\usepackage{amsfonts}` no preâmbulo.

Capítulo 4

Tabelas

Os ambientes *tabbing* e *tabular* são os que permitem alinhar o texto em colunas, mas há algumas diferenças entre eles:

- O ambiente *tabbing* pode ser usado somente no modo de texto; e *tabular* pode ser usado em qualquer modo (matemático, texto ...).
- O T_EX inicia uma nova página no meio do *tabbing*, mas não no meio do *tabular*.
- O T_EX determina automaticamente a altura e largura da tabela, enquanto no *tabbing* isso é decretado pelo usuário.

4.1 *Tabbing*

O ambiente *tabbing* é utilizado para tabular linhas, utilizando marcações para fazer referência às linhas e colunas que serão alinhadas.

Para marcar a posição das colunas, utiliza-se \=. Os comandos \> e \< avançam e recuam uma tabulação, respectivamente. Veja o exemplo a seguir.

Ex:

isto é um exemplo,	<code>\begin{tabbing}</code>
onde começa a próxima	isto é \= um exemplo, \\ <code>\> onde começa \= a próxima \\ palavra</code>
na última linha ?	<code>\> \> palavra \\ na última linha ?</code>
	<code>\end{tabbing}</code>

Note que \> fez avançar a 2^a linha até o 1^o marcador. Observe também que dois \> avançaram a 3^a linha em duas tabulações. O uso do \< será visto no próximo exemplo.

Pode ser conveniente que várias linhas sucessivas sejam tabuladas. Para isso, são usados os comandos \+ para avançar, e \- para recuar um bloco de linhas.

Ex:

```

\begin{tabbing}
1a linha com marcadores. 1a \= linha \= com \= marca\=dores. \+ \\
2a linha.                2a linha.\+ \\
3a linha.                3a linha.\+ \\
4a linha.                4a linha.\\
5a linha.                5a linha.\\
6a linha.                6a linha.\\
7a linha.                \> 7a linha.\\
8a linha.                \< 8a linha.\\
9a linha.                9a linha.\\
10a linha.               10a linha.\\
11a linha.               11a linha.\- \- \- \\
12a linha.                12a linha.
\end{tabbing}

```

Na 3ª linha, por exemplo, podemos observar que o comando `\+` avançou as tabulações seguintes até que um `\-` fosse utilizado na 11ª linha. No entanto, vemos que as linhas 7 e 8 não seguiram tal alinhamento por causa dos comandos `\>` e `\<`. Tais comandos não afetam as linhas de baixo, diferentemente de `\+` e `\-`, que mudam a tabulação das linhas seguintes.

É importante lembrar que o $\text{T}_{\text{E}}\text{X}$ considera um ou mais espaços como um único espaço. Portanto, vamos ver um exemplo de erro muito comum cometido neste ambiente:

```

\begin{tabbing}
coluna \= curta \\
coluna grande \> maior que curta
\end{tabbing}

```

```

coluna curta
coluna grande que curta

```

Pode ser visto que "maior que curta" ficou por cima de "grande". Por que será, já que foi dado espaço suficiente na linha de cima? Recordando: não importa quantos espaços sejam dados, só será considerado um. Uma dica para evitar que isso ocorra é colocar a maior palavra na primeira linha e eliminá-la usando `\kill`. Este comando faz com que a linha seja considerada na compilação, porém não apareça na tela.

```

Ex:
\begin{tabbing}
coluna grande \= maior que curta \kill
coluna \> curta \\
coluna grande \> maior que curta
\end{tabbing}
coluna curta
coluna grande maior que curta

```

4.2 Tabular

Este é semelhante ao `array`, diferindo deste pelo fato de poder ser usado em qualquer modo, não apenas no matemático.

Vamos ver um exemplo para entender os comandos.

```
\begin{tabular}{|c|l|rc|}
\hline
jan & fev & mar & abr \\ \hline
mai & jun & jul & ago \\ \cline{1-1} \cline{3-4}
set & out & nov & dez \\ \hline \hline
\end{tabular}
```

- As letras *c*, *l* e *r* significam centro, esquerda e direita respectivamente (center, left e right). Isto indica a posição do texto em relação a célula.
- As barras verticais (|) separando *c*, *l* e *r* são para fazer linhas verticais na tabela.
- \hline É para fazer linhas horizontais ao longo da tabela.
Repetindo várias vezes os mesmos comandos (| e \hline) seguidamente formam-se várias linhas.
- \cline{coli-colj} Faz linhas horizontais apenas entre as colunas i e j.
- & Divide os elementos de cada linha.
- \\ Indica o início de uma nova linha na tabela.

jan	fev	mar	abr
mai	jun	jul	ago
set	out	nov	dez

A largura da coluna pode ser determinada utilizando o comando `p{medida}`, ele deve ser colocado no lugar das letras *c*, *l* ou *r*

Ex:

```
\begin{tabular}{|l|c|p{3cm}|}\hline
col 1 & col 2 & coluna com 3 cm \\ \hline
col 1 & col 2 & podemos colocar uma frase nesta coluna
e ela será quebrada quando o tamanho
for maior que 3 cm. \\ \hline
isso não acontece nesta coluna ! & col 2 & \\ \hline
\end{tabular}
```

Faz:

col 1	col 2	coluna com 3 cm
col 1	col 2	podemos colocar uma frase nesta coluna e ela será quebrada quando o tamanho for maior que 3 cm.
isso não acontece nesta coluna !	col 2	

Há também a possibilidade de se montar uma tabela com multicolumnas, ou seja, uma célula grande pode ser construída com o agrupamento de células vizinhas em uma linha. É feito com o comando `\multicolumn{n}{pos}{item}`. Vamos ver o que isso significa na prática.

Primeiro, para a construção de uma tabela, é preciso saber o número máximo de colunas que ela terá para colocá-las no argumento situado após *tabular*. Conforme o comando citado acima, *n* é o número de colunas da tabela inteira que a multicolumna irá ocupar, *pos* é a posição que ficará o texto (*r / c*) e *item* é o texto que será digitado. É o mesmo que construir uma tabela dentro da outra.

Ex:

```
\begin{tabular}{|l|l|l|} \hline
segunda & \multicolumn{2}{|c|}{terça} \\ \hline
10 & 15 & 20 \\ \hline
15 & 10 & 25 \\ \hline
\end{tabular}
```

segunda	terça
10	15 20
15	10 25

A tabela é transformada em elemento flutuante, ou seja, ela será colocada pelo LaTeX no local de melhor visualização quando o ambiente *tabular* é colocado dentro do ambiente *table*.

Para definir o local da página em que a tabela ficará situada pode-se colocar:

- h**- Ficará onde foi digitado;
- b**- Ficará na parte inferior da página;
- t**- Ficará na parte superior da página;
- p**- Ficará em página separada.

Ex:

```
\begin{table}[b]
\begin{tabular}{|c|}
tabela
\end{tabular}
\end{table}
```

Entretanto, pode ser que o LaTeX não o aceite por motivos estéticos.

4.2.1 Tabelas Longas

Algumas vezes pode haver a necessidade de se usar grandes tabelas que ocupam mais que uma página, porém o ambiente *tabular* não permite que a tabela seja quebrada de acordo com a página.

O pacote *longtable* permite que isso seja feito através do ambiente *longtable*, que deve ser usado no lugar de *tabular*. Neste caso não é necessário utilizar o ambiente *table* para tornar a tabela em um elemento flutuante. Os comandos do ambiente *longtable* são os mesmos do *tabular*.

Lembrando que o pacote deve ser inserido no preâmbulo

```
\usepackage{longtable}
```

Ex:

```
\begin{longtable}{|c|c|}
```

```
\hline
  & & \\
  .
  .
  & & \\
\hline
\caption{Tabela longa.}
\end{longtable}
```

Capítulo 5

Movendo informações

Neste capítulo será visto como trabalhar com informações no documento através de referência cruzada, bibliografia, citação, inclusão de outros arquivos e figuras, sumário, etc..

5.1 Referência cruzada

Um dos motivos para as figuras, seções, equações e tabelas serem numeradas, é para posterior referência delas no texto. Por exemplo, escrevendo diretamente : "consulte a equação 10" pode ocorrer um problema, pois caso seja acrescentada futuramente outra equação antes dessa, seu número não será 10, mas sim 11. Logo, a referência estará errada. Para que isso não ocorra, pode-se criar um código para aquela equação com o comando `\label{código}` e referenciá-la com o comando `\ref{código}`.

Ex:

```
\begin{equation}
a^{2}+ b^{2}=c^{2} \label{equ:pitágoras}
\end{equation}
```

Consulte a `\ref{equ:pitágoras}` que é a equação de Pitágoras.

$$a^2 + b^2 = c^2 \tag{5.1}$$

Consulte a 5.1 que é a equação de Pitágoras.

Conforme forem sendo acrescentadas mais referências será preciso rodar o LaTeX mais de uma vez para que ele atualize a lista.

Também pode ser usado `\pageref{...}` para referência de páginas. E para referenciar partes do documento como capítulos e seções, basta fazer o mesmo que é feito com a equação.

Ex:

```
\section{Referência cruzada \label{sec:rc}}.
```

Uma dica para não se perder com a quantidade de códigos diferentes é identificar a referência com: equ:(equação), fig:(figura), tab:(tabela), teo:(teorema) etc..

É preciso compilar o LaTeX duas vezes: na primeira ele guarda a informação em um arquivo auxiliar e na segunda ele vai até este arquivo buscar esta informação e colocar no documento.

5.2 Dividindo o arquivo

Quando o arquivo fica muito grande é importante que ele seja dividido em arquivos menores, para que o tempo de processamento seja menor, e que no final seja reunido novamente. Isso é feito da seguinte forma:

```
\includeonly{lista de arquivos separados por
vírgula (sem .tex) no preâmbulo}
\include{arquivo (sem .tex) na ordem desejada}
```

Ex:

```
\includeonly{introducao,formatacao,capa}
. . .
\include{capa} ...
\include{formatacao} ...
\include{introducao} ...
```

É importante salientar a diferença entre os comandos `\includeonly` e `\include`. Vejamos um exemplo.

Vamos supor que o arquivo principal seja `apostila.tex` e que os capítulos da apostila estejam separados em outros arquivos `cap1.tex`, `cap2.tex` etc. Para incluí-los no texto, basta acrescentar `\include{cap1}` e quantos outros "includes" forem precisos. Agora, caso seja necessário compilar somente o arquivo `cap7.tex`, por exemplo, não convém compilar todos os outros uma vez que o tempo de processamento seria maior; assim, utiliza-se `\includeonly{cap7}` no preâmbulo.

É comum pensar na possibilidade de comentar as linhas referentes aos "includes" dos outros capítulos, desse modo, somente o `cap7.tex` seria compilado. Tal escolha não é recomendada pois quando se utiliza o `\include`, geram-se arquivos auxiliares (`.aux`) contendo informações sobre referências cruzadas e paginação, as quais seriam perdidas caso os arquivos não fossem criados. Então, quando se usa o `\includeonly`, é importante não apagar os arquivos auxiliares gerados anteriormente para que tais informações não sejam perdidas.

Usando `\input` no lugar de `\include`, o arquivo é inserido no meio da página onde foi colocado, ao contrário do `\include` que inicia uma nova página.

O LaTeX faz automaticamente o ajuste de numeração na hora em que reúne os arquivos.

Obs: Os arquivos que serão inseridos no documento não devem ter os comandos que aparecem no preâmbulo, nem `\begin{document}` e `\end{document}`.

5.3 Bibliografia

Uma das maneiras de fazer a bibliografia é utilizar o ambiente *thebibliography*. E cada referência começa com `\bibitem{livro}` e o comando `\cite{livro}` faz as referências no meio do documento.


```
\begin{thebibliography}{n}
  \bibitem{ref}{Livro}
\end{thebibliography}
```

No exemplo, n é o número máximo de itens de referência que terá o documento, ref é o código de referência do livro e $livro$ são os dados do livro. Este ambiente deve ser colocado no final do documento.

BibTeX

A outra maneira é através do utilitário BibTeX. Ele permite que sejam montados dados bibliográficos para posterior uso em seus documentos.

É usado `\cite` para citações no meio do texto, mas em vez de digitar a lista diretamente no documento, é usado o comando `\bibliography{nome}` com o nome dos arquivos que contêm o banco de dados.

Antes de saber como fazer isso, vamos ver como o LaTeX e o BibTeX interagem. Quando o documento é compilado pelo *latex*, é criado um arquivo com extensão `.aux` que contém todas as informações de referência cruzada. Quando seu documento tiver os comandos `\bibliography` e `\bibliographystyle`, este arquivo `*.aux` guardará as informações de citações e argumentos deste comando. Em seguida, quando o documento é compilado pelo *bibtex*, todas essas informações são lidas e é criado um novo arquivo com extensão `.bbl` contendo os comandos que produziram a lista. A próxima vez em que o documento for compilado pelo *latex*, o comando `\bibliography` lê o arquivo `*.bbl` e gera a lista bibliográfica.

Como fazer:

- Primeiramente deve-se escrever um arquivo (é importante ressaltar que este é um arquivo separado do documento principal) contendo os dados bibliográficos baseado nos tipos pré-definidos pelo LaTeX. Os principais são:

article São os artigos de jornais ou revistas.

book Um livro.

inbook Parte de um livro (capítulo, páginas etc.).

manual Documento técnico.

Os principais campos que devem ser preenchidos são:

author Autor

title Título

year Ano

publisher Editora

address Endereço (cidade, estado...).

Consulte a referência [1] para encontrar mais tipos.

```
Tipo@{código de referência,
  AUTHOR= "nome",
  TITLE = " Título ",
  PUBLISHER = "Editora",
```

```
ADDRESS = "Endereço",
. . .
YEAR = "Ano" }
```

Geralmente as interfaces gráficas disponíveis para o trabalho com o LaTeX, já disponibilizam opções em uma forma completa, cabendo ao usuário somente o preenchimento dos campos. Consulte o manual do seu editor.

Salve este arquivo com extensão `.bib` na mesma pasta onde está o documento;

- Insira o comando `\bibliographystyle{estilo}` no preâmbulo do documento e o comando `\bibliography{arquivo (sem .bib)}` no local onde deverá aparecer a bibliografia.

Os estilos podem ser:

plain É o mais usado. As entradas são colocadas em ordem alfabética.

unsrt As entradas aparecem na ordem de citação no meio do texto.

abbrv Semelhante aos anteriores, mas vem com nomes abreviados.

Obs: Só aparecerão na lista bibliográfica as referências que foram citadas. Para que elas apareçam na lista sem precisar que sejam citadas no texto, use o comando:

```
\nocite{nome dos códigos separados por vírgula}.
```

- Para gerar a bibliografia deve-se compilar o *latex*, em seguida o *bibtex* e o *latex* novamente.

Por exemplo:

É criado um arquivo contendo a descrição da referência, como feito abaixo:

```
@book{ apos:tex,

author = {Programa de Educação Tutorial},

title = {Apostila de \LaTeX},

publisher = {Universidade Federal Fluminense},

address = {Niterói - Rio de Janeiro,

year = {2004}
}
```

O arquivo é então salvo na mesma pasta com um nome qualquer e extensão `.bib`. Ex: `livros.bib`

É inserido no preâmbulo do documento o comando: `\bibliographystyle{estilo}`. E no fim do documento o comando: `\bibliography{arquivo (sem .bib)}`.

Ex:

```

\bibliographystyle{plain}
.
\begin{document}
.
.
\bibliography{livros}
\end{document}

```

Feito isso, é só rodar o *latex - bibtex - latex*.

5.4 Figuras

Para inserir figuras deve-se colocar no preâmbulo o pacote *graphicx* e depois usar o comando que permite inserir figura. Inicialmente, o formato de figura que o LaTeX aceitava era somente *EPS (Encapsulated PostScript)*, por ser mais usado. Porém, a necessidade de inclusão de figuras com formatos diferentes era grande, então, alguns usuários do LaTeX criaram pacotes que permitiam a inserção de outros formatos de figuras além do *.eps*. É usado o ambiente *figure* para poder tornar a figura um elemento flutuante, dar um título à figura e usar um código para referência.

```

\begin{figure}[argumento de posição como no table]
\includegraphics[medidas]{nome do arquivo}
\caption{Título da figura.}
\label{código de referência}
\end{figure}

```

As *medidas* são os parâmetros:

- *width* Largura;
- *height* Altura;
- *angle* Rotaciona a figura no sentido anti-horário.
- *scale* Muda a escala da figura.

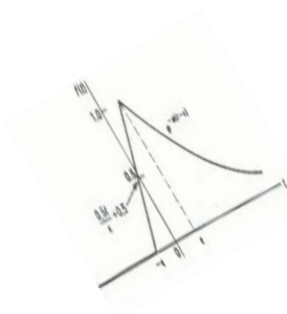


Figura 5.1: Gráfico.

```

\usepackage{graphicx}
...
\begin{document}
...
\begin{figure}[h]
  \centering % este comando é usado para centralizar a figura
  \includegraphics[width=4cm, height=6cm, angle=30]{grafico.jpg}
  \caption{Gráfico.}
  \label{fig:exem}
\end{figure}
...
\end{document}

```

Figuras .pdf Se o pacote *graphicx* for usado com o opcional [*pdftex*], torna-se possível inserir figuras no formato *.pdf. Neste caso, o documento não poderá ser compilado com o *latex* e sim com o *pdflatex*. Deve-se conferir se seu sistema oferece este recurso.

Ex:

```

\usepackage[pdftex]{graphicx}
...
\begin{document}
...
\begin{figure}[h]
  \centering
  \includegraphics{grafico.pdf}
  \caption{Gráfico.}
  \label{fig:exem}
\end{figure}
...
\end{document}

```

Compilando com o *pdflatex* é gerado diretamente o documento no formato .pdf ao invés de .dvi.

Figuras .jpg, .png, .pdf Para inserir figuras nestes formatos deve-se em primeiro lugar inserir a opção *dvipdfm* como argumento da classe do documento.

```
\documentclass[dvipdfm]{report}
```

Para inserir a figura é necessário convertê-la do formato original (.jpg, .png, .pdf) para o formato .bb (*bounding box*). Como fazer?

Suponha que o nome do arquivo seja *grafico.jpg*. Para transformá-lo em *gráfico.bb* basta abrir o *Command Prompt* e digitar:

```
ebb grafico.jpg
```

Isto transforma o arquivo de .jpg em .bb.

Obs: A figura não aparecerá no documento .dvi. É preciso convertê-la para .pdf. Para fazer isso, abra o *Command Prompt* e digite:

```
dvipdfm documento.dvi
```

Supondo que o nome do arquivo seja *documento*.

5.4.1 Subfiguras

Existe um pacote chamado *subfigure* que permite ao ambiente *figure* possuir mais de uma figura.

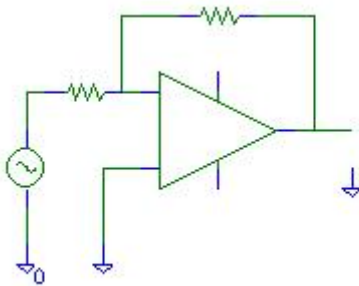
Antes de qualquer coisa deve-se declarar no preâmbulo o comando:

```
\usepackage{subfigure}
```

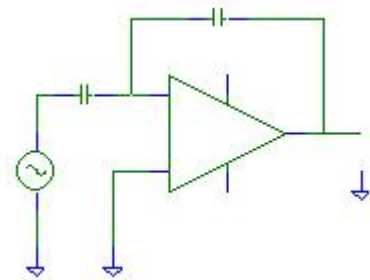
Veja o exemplo abaixo para entender o uso do comando.

Ex:

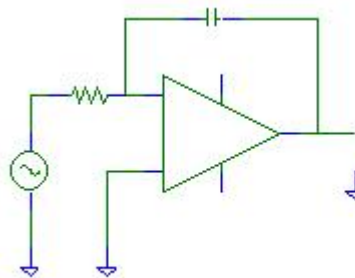
```
\begin{figure}[h]
  \subfigure[Primeira\label{fig:pri}]{\includegraphics{fig1.jpg}}
  \subfigure[Segunda\label{fig:seg}]{\includegraphics{fig2.jpg}}
  \center{\subfigure[Terceira\label{fig:ter}]{\includegraphics{fig3.jpg}}}
  \caption{Conjunto de figuras.}
  \label{fig:conj}
\end{figure}
```



(a) Primeira



(b) Segunda



(c) Terceira

Figura 5.2: Conjunto de figuras.

O título de cada uma das figuras 5.2 \ref{fig:conj} é colocado entre colchetes, assim como seu código de referência. A terceira figura 5.2(c) \ref{fig:ter} está dentro do comando `\center` para ficar centralizada.

5.5 Sumário

O sumário é feito facilmente através de um único comando: `\tableofcontents`, que deve ser colocado logo após `\begin{document}`, e o sumário é gerado automaticamente.

Ex:

```
...
\begin{document}
\tableofcontents
```

....

Caso haja alguma seção, figura ou tabela sem a numeração (por exemplo, quando é colocado o nome da seção com asterisco: `\section*`. Veja: 2.2.13) é possível que ela apareça no sumário usando o comando `\addcontentsline{arquivo}{seção}{nome}`, onde *arquivo* é a extensão da lista que deverá entrar (veja 5.7), *seção* é o título da seção ou capítulo, e *nome* é o nome que aparecerá na lista, podendo ser o mesmo título.

Ex:

```
\tableofcontents
...
\addcontentsline{toc}{chapter}{Prefácio}
```

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

É preciso compilar o arquivo duas vezes para que seja visualizado o sumário a cada mudança.

5.6 Índice

Uma das maneiras de produzir o índice remissivo é utilizando o ambiente *theindex* onde cada entrada é feita usando `\item`, a subentrada usando `\subitem` e a subsubentrada usando `\subsubitem`. Este ambiente produz o índice em duas colunas. O comando `\indexspace` faz um espaço vertical entre as entradas.

Ex:

```
\item babel 14

\item Color 44

    \subitem \verb=\=textcolor 56, 32

        \subsubitem color 45
\indexspace

\item article 15
```

Faz:

```
babel 14
Color 44
  \textcolor 56, 32
    color 45
```

```
article 15
```

O índice remissivo também pode ser criado facilmente da seguinte forma:

O programa *MakeIndex* Este é um programa que faz índice em um documento gerado pelo LaTeX.

Primeiro coloque no preâmbulo o pacote *makeidx* e o comando `\makeindex` e, no local onde deverá aparecer o índice, coloque o comando `\printindex`.

Para marcar os itens que apareceram no índice, use o comando `\index{item}`, onde *item* é a palavra que aparecerá no índice (*entrada*). Isso faz aparecer o item ao lado da página onde ele está localizado. *Subentradas* também podem ser geradas da seguinte forma: `\index{item!item!item}`. O ponto de exclamação serve para separar as entradas das subentradas.

Ex:

```
\documentclass[a4,12pt,oneside]{article}
\usepackage[T1]{fontenc}
\usepackage[brazil]{babel}
\usepackage{makeidx}
\makeindex
\begin{document}
...

\index{babel}
...
\index{Color!\verb=\=textcolor!color}
...
\index{Classe!book!article}

\printindex
\end{document}
```

Usando o exemplo acima, quando impresso, deverá aparecer no índice remissivo o seguinte:

```
babel, 12
```

```
Classe
  book
    report, 7
```

```
Color
  \textcolor
    color, 30
```

Para gerar o índice é preciso:

- Compilar o documento pelo *latex* para que ele gere um arquivo com extensão *.idx*;
- Em seguida compilar pelo *makeindex* com o nome do arquivo com extensão *.idx* para que ele gere um novo arquivo com extensão *.ind*;
- E compilar pelo *latex* novamente.

Sempre que houver uma mudança, deve-se seguir estas etapas de compilação.

5.7 Tipos de arquivos

Quando o documento é compilado, vários arquivos serão criados com o mesmo nome do documento. A lista abaixo dá uma breve explicação do significado de cada extensão de arquivo.

.tex Este é o arquivo principal, onde está o código fonte escrito.

.dvi Este é o arquivo pronto para ser visualizado e transformado em *ps* para imprimir. É o arquivo independente de dispositivo (*device independent*).

.aux É onde estão localizadas as informações de referência cruzada.

.toc Armazena os títulos das seções (*Table of contents*).

.lof Armazena os títulos das figuras (*List of figures*).

.lot Armazena os títulos das tabelas (*List of tables*).

.idx Contém o índice remissivo.

.log É o relatório da compilação feita, com os erros.

.bbl É o arquivo escrito pelo BibTeX.

Capítulo 6

Estrutura visual

Neste capítulo serão vistos conceitos básicos sobre como modificar a formatação padrão.

6.1 Cabeçalho e Rodapé da página

A página é formada por 3 partes: cabeçalho (topo da página), corpo (o texto) e rodapé (parte inferior da página). As características do cabeçalho e do rodapé da página são definidas pelo comando `\pagestyle`, enquanto a numeração é definida pelo comando `\pagenumbering`.

Os quatro estilos principais da página são feitos usando o comando `\pagestyle{estilos}`. Tais estilos são:

plain A página possui apenas numeração no rodapé;

empty Produz cabeçalho e rodapé vazios, sem numeração;

headings Depende do estilo do documento.

A página da frente (ímpar) vem com a seção, no caso de *book*, e subseção, no caso de *article*. Quando a página estiver definida como *oneside*, aparecerá nas páginas o capítulo, no caso de *book*, e seção, no caso de *article*. Caso não haja seções, só aparece a numeração no cabeçalho;

myheadings É especificado pelo usuário o que estará escrito no topo com os comandos:

- **markboth** – Quando o documento estiver em *twoside*:
`\markboth{página par}{página ímpar};`
- **markright** – Quando o documento estiver em *oneside*:
`\markright{páginas pares e ímpares}.`

Nos locais onde estão escritos página par e/ou ímpar, deverão ser escritas as palavras que aparecerão no cabeçalho.

A numeração da página é feita automaticamente com algarismos árabes, mas, para mudá-los, use o comando:

`\pagenumbering{roman}` Os números são colocados em romano;

Roman em romano maiúsculo;

alph em letras comuns e

Alph em letras comuns maiúsculas.

Se `\pagenumbering` for colocado no meio do texto, a numeração a partir daí será iniciada novamente com o novo estilo de numeração declarado.

O pacote *fancyhdr* Os comandos que foram vistos acima são úteis, porém limitados. Um pacote que adiciona recursos ao estilo da página é o *fancyhdr*.

Com ele pode-se, entre outras coisas:

- Dividir o rodapé e cabeçalho em 3 partes diferentes,
- Inserir linhas.

Em primeiro lugar deve-se inserir o pacote através do comando: `\usepackage{fancyhdr}` no preâmbulo, no caso do \LaTeX 2_ε. E no caso do \LaTeX 2.09 insira *fancyhdr* como um argumento opcional da classe do documento: `\documentstyle[fancyhdr,12pt,a4]{book}`

Após isso, defina no preâmbulo o estilo da página através do comando `\pagestyle{fancy}`.

Para páginas ímpares (ODD)

RO esquerda	CO (cabeçalho)centro	LO direita
corpo da página		
RO esquerda	CO (rodapé) centro	LO direita

Para páginas pares (EVEN)

RE esquerda	CE (cabeçalho)centro	LE direita
corpo da página		
RE esquerda	CE (rodapé) centro	LE direita

Tabela 6.1: Cabeçalho e rodapé.

Cada campo é definido pelos seguintes comandos:

- `\fancyhead[parâmetro]{informação}`
- `\fancyfoot[parâmetro]{informação}`

O usuário pode definir como *informação* o que quiser, ou colocar também o comando `\thepage` que diz onde o número da página vai aparecer.

O parâmetro é a indicação do campo em que informação irá aparecer. Por exemplo, observe a tabela 6.1. As letras O e E dizem respeito às páginas ímpares e pares, respectivamente. C, R e L, significam centro, direita e esquerda, respectivamente. Através de cada combinação dessas letras, fica-se especificado onde a informação estará.

Utilizando os comandos:

`\renewcommand{\headrulewidth}{medida}` e `\renewcommand{\footrulewidth}{medida}` é possível estabelecer a largura das linhas do cabeçalho e rodapé, respectivamente, através da medida dada.

Caso não seja usado nenhum desses comandos, o padrão que será gerado será:

Para páginas ímpares (ODD)

Capítulo	Seção
corpo da página	
Número da página	

Para páginas pares (EVEN)

Seção	Capítulo
corpo da página	
Número da página	

Este padrão é produzido da seguinte forma:

```
\fancyhead[LE,RO]{\rightmark}
\fancyhead[LO,RE]{\leftmark}
\fancyfoot[C]{\thepage}
```

Repare que `\rightmark` está se referindo à seção e `\leftmark`, ao capítulo. Caso o usuário queira, estes comandos também podem servir como *informação*.

O exemplo abaixo mostra como foi feito o cabeçalho desta apostila:

```
\pagestyle{fancy}
\fancyhead[LO,LE]{\it\nouppercase\leftmark}
\fancyhead[RO,RE]{\it\nouppercase\rightmark}
\fancyfoot[LO,LE]{\textsc{\uff}}
\fancyfoot[RO,RE]{\pet}
\fancyfoot[CO,CE]{\thepage}
\renewcommand{\footrulewidth}{0.4pt}
\renewcommand{\headrulewidth}{0.4pt}
```

O comando `\nouppercase` utilizado acima tem a função de colocar os capítulos e seções em letras minúsculas. Já os comandos `\uff` e `\pet` foram criados especialmente para uso nesta apostila. Para informações sobre novos comandos, consulte 6.7.

6.2 Área de impressão

A página em qualquer documento em LaTeX possui uma determinada configuração que depende de vários parâmetros, isto é, comandos que agem em determinadas partes da página. A figura 6.1 mostra todos esses comandos.

Cada tipo de classe de documento e papel escolhido tem um tamanho padrão para a impressão. Mas, é possível mudar esse tamanho. Uma das maneiras é através dos comandos:

`\addtolength{padrão}{medida}` – Adicionará ao padrão de medida do documento a medida que for escolhida, e;

`\setlength{padrão}{medida}` – Fixa a medida escolhida para o padrão. O comando anterior adiciona uma medida enquanto esse dá uma nova medida.

padrão são os comandos:

- `\textwidth` A largura padrão do texto na página;
- `\textheight` A altura padrão do texto na página;
- `\columnsep` A largura do espaço entre as colunas quando *twocolumn* estiver como opção de estilo;
- `\columnseprule` A largura da linha vertical colocada entre as colunas do texto usando *twocolumn*. O padrão é largura zero, por isso normalmente não aparece a linha;
- `\oddsidemargin` É a distância da borda esquerda do papel para a margem esquerda do texto menos uma polegada, em páginas ímpares quando *twoside* estiver declarado;
- `\evensidemargin` É o mesmo que `\oddsidemargin` só que para páginas pares;
- `\marginparwidth` Largura das notas marginais;
- `\topmargin` A distância da margem superior do papel para o topo do cabeçalho da página menos uma polegada;
- `\headheight` A altura da caixa que contém o cabeçalho.

E a *medida* pode ser colocada em centímetros ou em outra medida citada na seção 6.3.

Ex: `\addtolength{\textheight}{3.0cm}` adicionará 3 centímetros à altura da área de impressão da página.

Este comando deve ser colocado no preâmbulo.

6.3 Espaços e Medidas

Através dos seguintes comandos pode ser dado espaçamento entre linhas e palavras:

- `\hspace{medida}` - Adiciona espaço entre as palavras, onde a medida pode ser em:
 - (cm) Centímetros;
 - (in) Polegada ($1in = 2.54cm$);
 - (pt) Ponto ($1in = 72.27pt$);

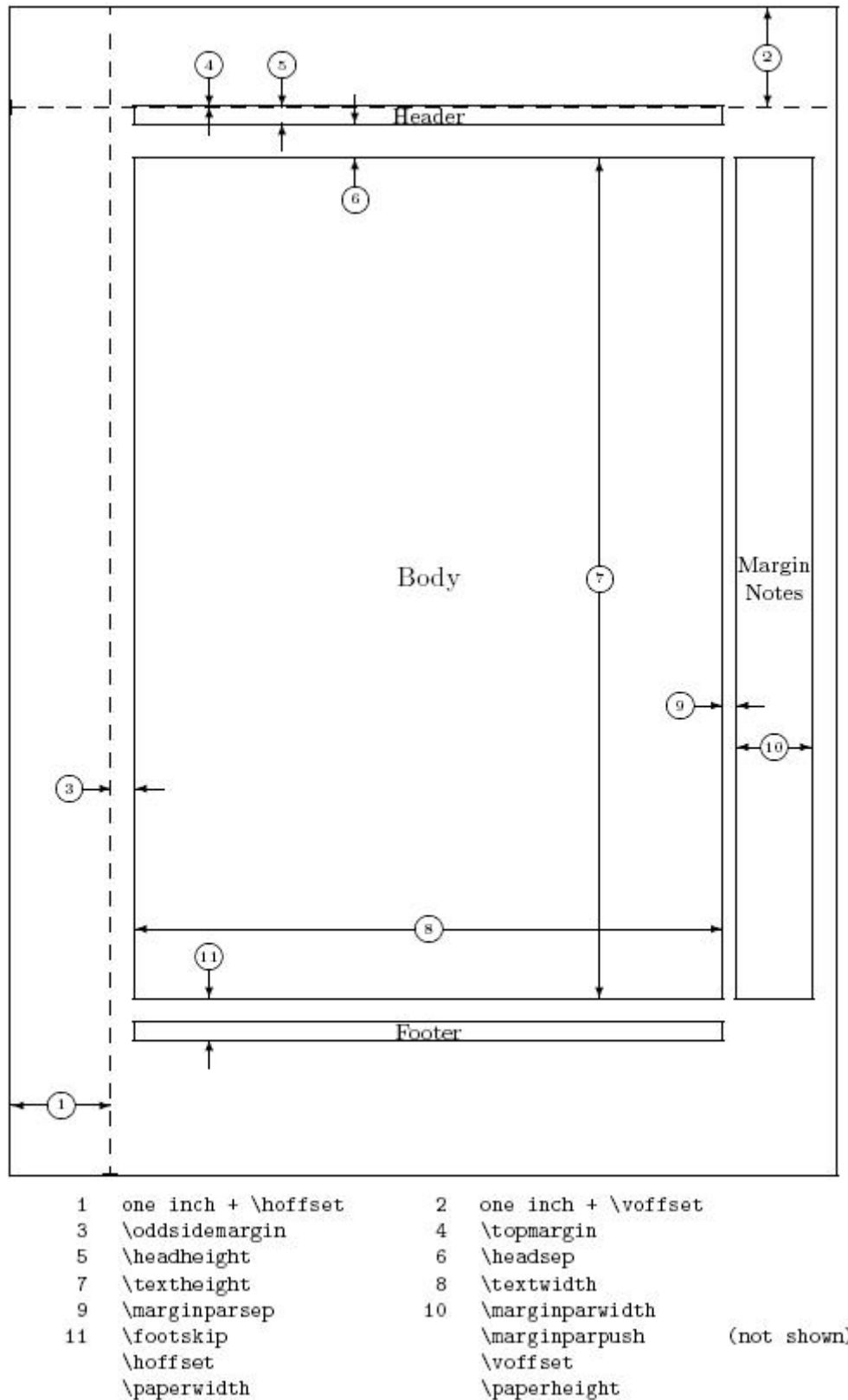


Figura 6.1: Medidas da página.

6.4 Caixas

Usando o comando `\fbox{palavra}`, é construída uma caixa ao redor da palavra `Exemplo`, e através do comando `\framebox[medida][posição]{palavra}` pode-se controlar o tamanho da caixa.

Ex: `texto \framebox[15ex][r]{palavra} texto.`

texto palavra texto.

Um erro comum é deixar um espaço menor para a caixa do que o tamanho do texto.

Ex: `texto \framebox[5ex][l]{palavra} texto.`

texto palavra texto.

No exemplo, verifica-se que o texto fica, em parte, fora da caixa. Além disso, os textos ficam sobrepostos pois o alinhamento das palavras é feito de acordo com o tamanho da caixa, e não com o texto dentro dela.

Usando o comando `\makebox` da mesma forma que `\framebox` também é criada uma caixa, só que sem linha.

Ex: `texto \makebox[6ex][r]{palavra} texto.`

texto palavra texto.

O comando `\raisebox{altura}{palavra}` faz uma caixa onde a *palavra* é elevada a uma *altura* escolhida, podendo também ser uma medida negativa (para baixo -).

Ex: `texto \raisebox{1.0ex}{palavra} texto.`

texto palavra texto.

6.5 Cores

Primeiramente, para usar cor é preciso colocar no preâmbulo o comando `\usepackage{color}`. Isso é para o LaTeX carregar o pacote *color* e reconhecer os comandos de cores. Veja alguns comandos:

- `\definecolor{nome}{modelo}{parâmetro}` - Este define a cor, onde *nome* é o nome da cor, *modelo* é o modelo da cor (com o principal sendo *rgb*: red, green, blue) e *parâmetro* é o código da cor segundo o modelo usado.

Ex:

`\definecolor{azul}{rgb}{0,0,1}` define a cor azul.

O padrão *rgb* é o mais usado devido ao seu grande número de combinações de cores feitas com os códigos: peso da cor vermelha, peso da cor verde, peso da cor azul, onde os números variam entre 0 e 1. Lembrando que o separador decimal é o ponto(.), não a vírgula. Ex: escreve-se *0.71* ao invés de *0,71*.

- `\textcolor{cor}{palavra}` - Muda cor da palavra selecionada. A cor pode ser escrita diretamente em inglês ou usar o *definecolor*, definindo o nome da cor em português.

Ex:

```
\definecolor{verdemar}{rgb}{0.59,0.78,0.65}
\textcolor{verdemar}{texto em cor personalizada}
texto em cor personalizada
```

- `\color{cor}` - Muda a cor do texto inteiro. Para mudar apenas um trecho do texto use chaves neste trecho.

Ex:

```
{\color{blue}{trecho do texto}}
trecho do texto
```

- `\pagecolor{cor}` - Muda a cor do fundo da página.
- `\colorbox{cor}{texto}` Gera uma caixa com o fundo da cor que foi escolhida. Para fazer uma caixa em um grande pedaço de texto use *minipage*, veja seção 6.6.

Ex:

```
\colorbox{red}{palavra}
palavra
```

- `\fcolorbox{bcor}{ccor}{texto}` - Gera uma caixa com cor *ccor* e borda *bcor*.

Ex:

```
\fcolorbox{verdemar}{green}{palavra}
palavra
```

Importante: dependendo do visualizador de DVI as cores não serão vistas, mas se converter em PS ou PDF, aparecerão normalmente.

6.6 Minipage

Esse é um ambiente que cria uma área com formato de uma página, com largura desejada através do comando:

```
\begin{minipage}[posição t ou b]{largura}
  texto
\end{minipage}
```


O texto pode conter outros ambientes, podendo colocar até notas de pé de página ^a, mas esta nota não aparecerá no fim da página comum. Há também possibilidade de se colocar moldura com o comando `\fbox{ambiente minipage}` e também trabalhar com os comandos de cores sem problemas, através de combinações. Os argumentos *b* e *t* permitem o alinhamento do topo (*t*) e do fim (*b*) da *minipage* em relação a linha do texto. Devem ser usados quando há uma outra *minipage* do lado.

^aesta nota aparece no fim da minipágina

```
\begin{minipage}[t]{0.5\textwidth}
  O texto pode conter (...) notas de pé de página
  \footnote{esta nota aparece (...) minipágina}
  ... do lado.
\end{minipage}
```

Outra possibilidade de colocar moldura é usando *tabular*, onde os itens serão as minipáginas.

6.7 Novos comandos e ambientes

O \LaTeX também permite que seja modificado o nome de algum comando para o nome escolhido ou que seja criado um macro, ou seja, um comando que sintetize outros comandos. Isso é feito através de `\newcommand{novos comando}{definição}`

Ex: Se uma mesma frase for utilizada várias vezes ao longo do texto, é útil criar um comando que a resumisse. Então, querendo digitar: Universidade Federal Fluminense, coloque-se no preâmbulo

```
\newcommand{\uff}{Universidade Federal Fluminense}. Depois é só usar \uff para aparecer a frase: Universidade Federal Fluminense.
```

Pode-se também montar um comando que tenha uma estrutura em que os argumentos variem.

Ex:

```
\newcommand{\vt}[3]{\emph{vetor}$(#1;#2;#3)$}
os vetores \vt{5x}{3x}{7x} e \vt{9w}{8w}{3w} são ...
```

Faz:

os vetores $\text{vetor}(5x; 3x; 7x)$ e $\text{vetor}(9w; 8w; 3w)$ são ...

Vamos ver o que significa cada coisa:

- `\vt` É o nome dado ao novo comando.
- O `[3]` é o número de argumentos que variam; no caso, as 3 coordenadas.
- `#1;#2;#3` Indica o local em que aparecerão os argumentos.

- `\vt{}`... é o uso do comando no qual os argumentos são colocados entre chaves.

Para fazer ambientes há uma pequena diferença:

```
\newenvironment{emphit}{\begin{itemize}\em}{\end{itemize}}
\begin{emphit}
\item este é o novo item enfatizado
\end{emphit}
```

Faz:

- *este é o novo item enfatizado*

O que foi feito:

- `emphit` É o nome do novo ambiente.
- `itemize` É o ambiente base, pois geralmente os novos ambientes são feitos a partir de um existente.
- `\em` Faz o texto do item ficar enfatizado.

Também podem ser definidos ambientes com argumentos variáveis, assim como comandos.

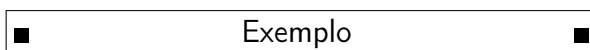
Ex: No preâmbulo colocando:

```
\newenvironment{meuambi}[1]
{\begin{center}
\fbbox{\rule{1ex}{1ex}\hspace{15ex}{#1}\hspace{15ex}
\rule{1ex}{1ex}}
{\end{center}}}
```

E no meio do texto:

```
\begin{meuambi}
{Exemplo}
\end{meuambi}
```

Será visto o seguinte ambiente:



Apêndice A

Utilizando o LaTeX através de um Terminal de Comando

Geralmente, os usuários costumam utilizar o sistema LaTeX através de editores de textos específicos e que já possuem em sua interface os comandos de compilação através de ícones, bastando clicá-los para que se tenha os arquivos gerados em `.dvi`, `.pdf`, etc.. Porém, há casos em que é necessário utilizar o LaTeX diretamente através de um terminal de comando.

Este capítulo tem a intenção de mostrar ao usuário alguns comandos básicos para o completo trabalho com o LaTeX.

Suponha que o arquivo principal chame-se `arquivo.tex` e que o arquivo que contenha a lista bibliográfica se chame `refer.bib`.

DVI Para compilar o arquivo e gerar um documento em `.dvi`, digite:

```
latex arquivo.tex
```

PS Para transformar o `.dvi` em `.ps`, digite:

```
dvips arquivo.dvi
```

PDF Para transformar o `.dvi` em `.pdf`, digite:

```
dvi2pdf arquivo.dvi
```

Para gerar/transformar o documento de `.tex` diretamente para `.pdf`, compile o arquivo digitando:

```
pdflatex arquivo.tex
```

BIBTEX Para gerar o arquivo em .dvi contendo a lista bibliográfica do arquivo .bib, digite:

```
latex arquivo.tex
```

```
bibtex refer.tex
```

```
latex arquivo.tex
```

MAKEINDEX Para gerar o arquivo em .dvi contendo índice remissivo utilizando o pacote *makeidx*, digite:

```
latex arquivo.tex
```

```
makeindex arquivo.idx
```

```
latex arquivo.tex
```

Apêndice B

Símbolos matemáticos

Todos esses símbolos são usados apenas em ambientes matemáticos, portanto, para inserí-los no meio de um texto, use \$... \$.

lim	<code>\lim</code>	arg	<code>\arg</code>	cot	<code>\cot</code>
lim inf	<code>\liminf</code>	ker	<code>\cos</code>	coth	<code>\coth</code>
arccos	<code>\arccos</code>	lg	<code>\lg</code>	max	<code>\max</code>
arcsin	<code>\arcsin</code>	cosh	<code>\cosh</code>	csc	<code>\csc</code>
arctan	<code>\arctan</code>	ln	<code>\ln</code>	min	<code>\min</code>
det	<code>\det</code>	exp	<code>\exp</code>	hom	<code>\hom</code>
sec	<code>\sec</code>	sinh	<code>\sinh</code>	tan	<code>\tan</code>
dim	<code>\dim</code>	gcd	<code>\gcd</code>	inf	<code>\inf</code>
sin	<code>\sin</code>	sup	<code>\sup</code>	tanh	<code>\tanh</code>

Tabela B.1: Funções.

<code>^</code>	<code>\hat{}</code>	<code>ˇ</code>	<code>\check{}</code>	<code>˘</code>	<code>\breve{}</code>	<code>´</code>	<code>\acute{}</code>
<code>`</code>	<code>\grave{}</code>	<code>˜</code>	<code>\tilde{}</code>	<code>-</code>	<code>\bar{}</code>	<code>→</code>	<code>\vec{}</code>
<code>·</code>	<code>\dot{}</code>	<code>¨</code>	<code>\ddot{}</code>				

Tabela B.2: Acentos matemáticos.

\leftarrow	<code>\leftarrow</code>	\longleftarrow	<code>\longleftarrow</code>	\uparrow	<code>\uparrow</code>
\Lleftarrow	<code>\Lleftarrow</code>	\Llongleftarrow	<code>\Llongleftarrow</code>	\Uparrow	<code>\Uparrow</code>
\rightarrow	<code>\rightarrow</code>	\longrightarrow	<code>\longrightarrow</code>	\downarrow	<code>\downarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>	\Rlongrightarrow	<code>\Rlongrightarrow</code>	\Downarrow	<code>\Downarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>	\updownarrow	<code>\updownarrow</code>
\Lleftrightarrow	<code>\Lleftrightarrow</code>	\Llongleftrightarrow	<code>\Llongleftrightarrow</code>	\Updownarrow	<code>\Updownarrow</code>
\mapsto	<code>\mapsto</code>	\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\hookrightarrow	<code>\hookrightarrow</code>	\hookrightarrow	<code>\hookrightarrow</code>	\searrow	<code>\searrow</code>
\leftharpoonup	<code>\leftharpoonup</code>	\rightharpoonup	<code>\rightharpoonup</code>	\swarrow	<code>\swarrow</code>
\leftharpoondown	<code>\leftharpoondown</code>	\rightharpoondown	<code>\rightharpoondown</code>	\nwarrow	<code>\nwarrow</code>
\rightleftharpoons	<code>\rightleftharpoons</code>				

Tabela B.3: Setas.

\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>	\models	<code>\models</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>	\perp	<code>\perp</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>	\mid	<code>\mid</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>	\parallel	<code>\parallel</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>	\bowtie	<code>\bowtie</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>	\smile	<code>\smile</code>
\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\neq	<code>\neq</code>	\frown	<code>\frown</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\doteq	<code>\doteq</code>		
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\propto	<code>\propto</code>		

Tabela B.4: Símbolos de relação.

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\diamond	<code>\diamond</code>	\oplus	<code>\oplus</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\triangleup	<code>\triangleup</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\uplus	<code>\uplus</code>	\triangledown	<code>\triangledown</code>	\otimes	<code>\otimes</code>
\div	<code>\div</code>	\sqcap	<code>\sqcap</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
$*$	<code>\ast</code>	\sqcup	<code>\sqcup</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\star	<code>\star</code>	\vee	<code>\vee</code>	\triangleleft	<code>\triangleleft</code>	\bigcirc	<code>\bigcirc</code>
\circ	<code>\circ</code>	\wedge	<code>\wedge</code>	\triangleright	<code>\triangleright</code>	\dagger	<code>\dagger</code>
\bullet	<code>\bullet</code>	\setminus	<code>\setminus</code>	\trianglelefteq	<code>\trianglelefteq</code>	\ddagger	<code>\ddagger</code>
\cdot	<code>\cdot</code>	\wr	<code>\wr</code>	\trianglerighteq	<code>\trianglerighteq</code>	\amalg	<code>\amalg</code>

Tabela B.5: Símbolos de operação binária.

Minúsculas

α	<code>\alpha</code>	θ	<code>\theta</code>	o	<code>o</code>	τ	<code>\tau</code>
β	<code>\beta</code>	ϑ	<code>\vartheta</code>	π	<code>\pi</code>	υ	<code>\upsilon</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	ϕ	<code>\phi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	φ	<code>\varphi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	χ	<code>\chi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ψ	<code>\psi</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>	ω	<code>\omega</code>
η	<code>\eta</code>	ξ	<code>\xi</code>				

Maíúsculas

Γ	<code>\Gamma</code>	Λ	<code>\Lambda</code>	Σ	<code>\Sigma</code>	Ψ	<code>\Psi</code>
Δ	<code>\Delta</code>	Ξ	<code>\Xi</code>	Υ	<code>\Upsilon</code>	Ω	<code>\Omega</code>
Θ	<code>\Theta</code>	Π	<code>\Pi</code>	Φ	<code>\Phi</code>		

Tabela B.6: Letras gregas.

\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>	∞	<code>\infty</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>	\square	<code>\square</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>	\diamond	<code>\diamond</code>
\jmath	<code>\jmath</code>	\surd	<code>\surd</code>	\flat	<code>\flat</code>	\triangle	<code>\triangle</code>
ℓ	<code>\ell</code>	\top	<code>\top</code>	\natural	<code>\natural</code>	\clubsuit	<code>\clubsuit</code>
\wp	<code>\wp</code>	\perp	<code>\perp</code>	\sharp	<code>\sharp</code>	\diamondsuit	<code>\diamondsuit</code>
\Re	<code>\Re</code>	\parallel	<code>\parallel</code>	\backslash	<code>\backslash</code>	\heartsuit	<code>\heartsuit</code>
\Im	<code>\Im</code>	\angle	<code>\angle</code>	∂	<code>\partial</code>	\spadesuit	<code>\spadesuit</code>
\mho	<code>\mho</code>	\sum	<code>\sum</code>	\prod	<code>\prod</code>	\coprod	<code>\coprod</code>
\int	<code>\int</code>	\oint	<code>\oint</code>	\bigcap	<code>\bigcap</code>	\bigcup	<code>\bigcup</code>
\bigsqcup	<code>\bigsqcup</code>	\bigvee	<code>\bigvee</code>	\bigwedge	<code>\bigwedge</code>	\bigodot	<code>\bigodot</code>
\bigotimes	<code>\bigotimes</code>	\bigoplus	<code>\bigoplus</code>	\biguplus	<code>\biguplus</code>	\dots	<code>\dots</code>
\dots	<code>\dots</code>	\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>		

Tabela B.7: Símbolos variados.

Apêndice C

Outros símbolos

Acentos

ò	\‘{o}	õ	\~{o}	ö	\v{o}	ç	\c{}
ó	\’{o}	ō	\o=	ő	\H{o}	˘	\d{}
ô	\^ {o}	ô	\. {o}	ô	\t{o}	˙	\b{}
ö	\" {o}	ö	\u{o}			˘	

Símbolos estrangeiros

œ	\oe	å	\aa	ł	\l	¿	?‘
Œ	\OE	Å	\AA	Ł	\L	¡	!‘
æ	\ae	ø	\o	ß	\ss		
Æ	\AE	Ø	\O				

Tabela C.1: Símbolos estrangeiros e acentos.

†	\dag	¶	\P	✓	\checkmark	¥	\yen
‡	\ddag	©	\copyright	✠	\maltese	&	\&
§	\S	£	\pounds	®	\circledR	%	\%
#	\#	—	_	\$	\\$	L ^A T _E X	\LaTeX
{	\{	}	\}	L ^A T _E X 2 _ε	\LaTeXe	T _E X	\TeX

Tabela C.2: Símbolos diversos.

Referências Bibliográficas

- [1] Lamport, Leslie, *LaTeX: A Document Preparation System, User's Guide & Reference Manual*, Addison-Wesley Publishing Company, 1986.
- [2] T. Oitker, H. Partl, I. Hyna, E. Schlegl, *The Not so short introduction to LaTeX 2_ε*, Tradução D. A. Polli, USP, 2000.
- [3] H. Kopka e P. W. Daly, *A guide to LaTeX 2_ε, Document preparation for beginners and advanced users*, Addison-Wesley Publishing Company, 1995.
- [4] **CTAN** (Comprehensive TeX Archive Network), www.ctan.org, Este site é referência mundial para materiais relacionados ao TeX e LaTeX.
- [5] **TeX-Br**, <http://biquinho.furg.br/tex-br/>, Página dos usuários brasileiros de TeX e LaTeX.