ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/343749499

Latex: Elaboração de Documentos Digitais

Book · August 2020

CITATIONS
0

READS
2,247

3 authors:

Alessandro Vivas Andrade
Federal University from Jequitinhonha and Mucuri's Valleys
94 PUBLICATIONS **222** CITATIONS

SEE PROFILE

Luciana Pereira de Assis
Federal University from Jequitinhonha and Mucuri's Valleys
91 PUBLICATIONS **215** CITATIONS

SEE PROFILE

Leonardo Araujo
Federal University of São João del-Rei
26 PUBLICATIONS **19** CITATIONS

SEE PROFILE

All content following this page was uploaded by Alessandro Vivas Andrade on 19 August 2020.

The user has requested enhancement of the downloaded file.

LaTeX

Elaboração de Documentos Digitais

Alessandro Vivas
Leonardo Araújo
Luciana Assis

1ª Edição

2020

Leonardo Carneiro Araújo, Alessandro Vivas Andrade,
Luciana Pereira de Assis

LATEX

Elaboração de Documentos Digitais

Diamantina
2020

EDITOR: Alessandro Vivas Andrade
PROJETO GRÁFICO: Alessandro Vivas e Leonardo Araújo
CAPA: Leonardo Araújo

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Andrade, Alessandro Vivas
Latex [livro eletrônico] : elaboração de documentos digitais / Alessandro Vivas Andrade, Luciana Pereira de Assis, Leonardo Carneiro de Araújo. -- 1. ed. -- Diamantina, MG : Alessandro Vivas Andrade, 2020.
PDF

ISBN 978-65-00-07614-1

1. Ciência da Computação 2. Computação gráfica 3. Editoração eletrônica 4. LateX (Programa de computador) 5. Software 6. Trabalhos científicos - Editoração I. Assis, Luciana Pereira de. II. Araújo, Leonardo Carneiro de. III. Título.

20-42160 CDD-686.22544416

Índices para catálogo sistemático:

1. Editoração eletrônica : Tecnologia 686.22544416

Maria Alice Ferreira - Bibliotecária - CRB-8/7964

ISBN: 978-65-00-07614-1

Prefácio da Primeira Edição

A principal motivação desta publicação é apresentar de uma maneira didática os principais conceitos do Editor LaTeX. Alunos de graduação e pós-graduação muitas vezes tem dificuldade de usar o LaTeX para produção de seus Trabalhos de Conclusão de Curso, Dissertação de Mestrado ou Tese Doutorado. A dificuldade inicial é normal mas, uma vez superada, o uso do LaTeX passa a ser simples, facilitando a criação de documentos (principalmente os longos e complexos) e proporcionando uma estética muito mais agradável.

Esta obra reuniu os conceitos básicos para a produção de trabalhos acadêmicos e científicos. Entretanto, o assunto não se esgota por aqui. Este material será revisado e continuaremos a editar esta publicação com o objetivo de reunir um material mais abrangente para auxiliar e incentivar os leitores para o uso do LaTeX. Recomendamos ainda que, sempre que necessário, busque mais informações e resolva as dúvidas buscando na internet e consultando a comunidade, através dos fóruns como o StackExchange e LaTeX.org, dentre outros. Grande parte das dúvidas já foram postas e possivelmente as respostas estão em algum lugar na internet, basta procurar.

Sobre os Autores

Alessandro Vivas Andrade é natural de Lavras/MG e atualmente reside em Diamantina/MG. É professor do Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) onde leciona as disciplinas de Redes de Computadores e Sistemas Distribuídos. Também atua como Professor do Curso de Mestrado em Programa de Pós-Graduação em Educação em Ciências, Matemática e Tecnologia (PPGECMAT/UFVJM). Graduou-se em Engenharia Elétrica (UFMG) e depois cursou Mestrado e Doutorado ambos em Engenharia Elétrica na UFMG. Tem interesses nas áreas de Inteligência Artificial, Aprendizado de Máquina e Ciência de Dados.

Luciana Pereira de Assis é natural de Belo Horizonte/MG e atualmente reside em Diamantina/MG. É professora do Curso de Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) onde leciona as disciplinas de Algoritmos e Estrutura de Dados, Pesquisa Operacional e Inteligência Artificial. Também atua como Professor do Curso de Mestrado em Gestão em Instituições de Ensino (PPGGIED/UFVJM). Graduou-se em Ciência da Computação (UNIBH) e depois cursou Mestrado em Ciência da Computação (UFMG) e Doutorado em Engenharia Elétrica na UFMG. Tem interesses nas áreas de Otimização, Inteligência Artificial e Análise de Redes Sociais.

Leonardo Carneiro de Araújo é natural de Belo Horizonte/MG. É professor da Universidade Federal de São João del Rei (UFSJ) onde leciona as disciplinas de Teoria da Informação, Processamento de Áudio e Vídeo, Análise de Sinais e Sistemas, dentre outras. Graduou-se em Engenharia Elétrica (UFMG), depois cursou Mestrado e Doutorado em Engenharia Elétrica (UFMG). Tem interesse nas áreas de Teoria da Informação, Linguística e Linguística Quantitativa, Reconhecimento de Fala, Reconhecimento de Padrões e Inteligência Artificial.

Conteúdo

1 Introdução **17**
1.1 Palavras iniciais 18

2 Instalação das Ferramentas **25**
2.1 Visão Geral da Instalação 26
2.2 Instalação do Compilador de LaTeX 26
2.2.1 Linux 26
2.2.2 Instalação do Latex do Windows 27
2.2.3 Instalação do LaTeX no Mac OS 28
2.3 Instalando os Editores 29
2.3.1 Instalação do TexMaker 29
2.4 Gerenciadores de Bibliografia 29

3 Primeiro Documento **31**
3.1 Sintaxe 32
3.2 Comentários 33
3.3 Codificação e suporte a diferentes línguas 34
3.3.1 Codificação de caracteres 35
3.4 Codificação de fonte 40
3.5 Anatomia do documento 41
3.6 Pacotes 44
3.7 Divisão do documento 45
3.8 Ambientes 46
3.9 Tipografia das seções 46
3.10 Um exemplo de documento mais completo 47

4 Texto **51**
4.1 Elemento básico de um documento 52
4.2 Formatação do texto 52

4.3 Espaçamento 53
4.4 Quebra de linha e página 56
4.5 Hifenização e agrupamentos 58
4.6 Alinhamento do texto 59
4.7 Citações e notas de rodapé 60
4.8 Acentos 61
4.9 Organização 62
4.10 Numeração 63

5 Trabalhando com Figuras 65
5.1 Ambiente `figure` 66
5.2 Opções de posicionamento 66
5.3 Rotacionando figuras 69
5.4 Alterando o altura, largura e escala 71
5.4.1 Alterando a largura 71
5.4.2 Alterando a altura 72
5.4.3 Alterando a escala 73
5.5 Cortando a figura 74
5.6 Opções conjuntas 75
5.7 Inserindo várias figuras 75

6 Tabelas 79
6.1 Tabelas 80
6.2 Elementos básicos 80
6.3 Tabelas longas 84

7 Listas 89
7.1 Introdução 90
7.2 Listas não numeradas simples 90
7.3 Listas não numeradas aninhadas 91
7.4 Listas numeradas 93
7.5 Listas com descrição 99
7.6 Listas aninhadas 100
7.7 Listas com várias colunas 101

8 Fórmulas Matemáticas 103
8.1 Formalismo matemático 104
8.2 Fórmulas matemáticas no LaTeX 104
8.2.1 Operadores 107
8.2.2 Parênteses, colchetes e chaves 110

8.2.3 Somatórios, integrais e produtórios 110
8.2.4 Matrizes 111
8.3 Teoremas e definições 113
8.4 Formatação de caracteres 114
8.5 Diacríticos 115
8.6 Lista de símbolos 116

9 Criando Índice Remissivo 121
9.1 Importando o pacote 122
9.2 Inclua os índices 122
9.3 Imprima o índice 122
9.4 Compilando 123
9.5 Exemplo completo 123

10 Bibliografia 125
10.1 Introdução 126
10.2 Anatomia do documento 126
10.3 O arquivo `.bib` 127
10.4 Gerando a bibliografia 128
10.5 Estilos 129
10.6 Softwares de gerenciamento 132

11 Listagens 133
11.1 Introdução 134
11.2 Listings 134
11.2.1 Inserindo Legenda 134
11.2.2 Criando Referência ao Código 135
11.2.3 Definindo a Linguagem 135
11.2.4 Configurações Gerais 135
11.2.5 Definindo Tamanho da Fonte 136
11.2.6 Comentários 136
11.2.7 Cor do Comandos 137
11.2.8 Numeração 138
11.2.9 Alterar o nome da Listagem 139
11.2.10 Quebra Automática de Linhas 140
11.2.11 Acentuação e Caracteres Especiais 141

12 Erros 143
12.1 Erros 144
12.2 Erros comuns 144

12.3 Solucionando erros 146

Lista de Figuras

1.1 Exemplo de um Documento em LaTeX 21
1.2 Comparativo entre LaTeX e outros Editores (figura adaptada de [10]). 23

2.1 Site do Miktex. 28
2.2 Site do MacTex. 28

3.1 Retrato do Marquês de Lafayette feito por Samuel Morse em 1825 (Wikimedia: domínio público). 36
3.2 Código Baudout [9]. 37
3.3 Código Baudot-Murray adaptado (ITA2). (`http://www.quadibloc.com/crypto/images/improved.gif`) 38
3.4 Tabela ASCII contida no manual TermiNet 300 da GE (`http://archive.computerhistory.org/resources/text/GE/GE.TermiNet300.1971.102646207.pdf`) 39
3.5 Exemplo de um documento simples. 43
3.6 Exemplo do documento. 50

4.1 Como o TeX faz a organização do texto utilizando caixas. . . 59

5.1 Legenda da Imagem 67
5.2 Opções de posicionamento de Figuras 68
5.3 Opções Rotação 70
5.4 Trabalhando com a Largura 71
5.5 Trabalhando com a altura 72
5.6 Trabalhando com a escala 73
5.7 Cortando a Figura 75
5.8 Opções Compostas 76
5.9 Várias Figuras 77

6.1 Exemplo de tabela longa gerada pela Listagem 6.3. 86
6.2 Exemplo de tabela longa gerada pela Listagem 6.4. 87

9.1 Índice Remissivo 124

11.1 Resultado da Listagem 11.7 137
11.2 Resultado da Listagem 11.8 137
11.3 Resultado da Listagem 11.9 138
11.4 Resultado da Listagem 11.10 139
11.5 Resultado da Listagem 11.12 140
11.6 Resultado da Listagem 11.13 142

Listagens

1.1 Exemplo de Documento em Latex 20

2.1 Instalação do Compilador Latex em distribuições Linux da família Debian. 26
2.2 Instalação Completa do LaTeX 27
2.3 Instalação do Texmaker no Linux. 29

3.1 Trabalhando com espacos. 32
3.2 Ambiente `equation`. 32
3.3 Sintaxe de comandos em LaTeX. 33
3.4 Utilização do comando `\textbf` para escrever em negrito. . . 33
3.5 Comentários. 33
3.6 Comentário em bloco utilizando `comment`. 34
3.7 Especificando a codificação do seu código `.tex`. 34
3.8 Especificando a língua do documento. 35
3.9 Especificando codificação e língua para um documento. 35
3.10 Especificando a codificação de fonte. Neste exemplo adotaremos a codificação Cork (T1). 41
3.11 Documento bem simples. 41
3.12 Documento simples. 42
3.13 Usando seções para organizar o documento. 45
3.14 Criando uma seção sem numeração. 45
3.15 Exemplo de um documento simples em LaTeX. 48

4.1 Alguns tipos de ênfase de fonte disponíveis. 52
4.2 Variando o tamanho do texto. 53
4.3 Fonte com tamanho arbitrário. 53
4.1 Comando para alterar o espaçamento do texto. 53
4.2 Comando para alterar o espaçamento do texto. 53
4.4 Adicionando espaços na horizontal e vertical. 54

4.3 Utilizando as unidades pré-definidas no LaTeX. 55
4.4 Fazendo o LaTeX ser menos meticuloso. 56
4.5 Outra forma de fazer o LaTeX ser menos meticuloso. 57
4.6 Comando para alterar o espaçamento do texto. 58
4.5 Mantendo um texto coeso. 58
4.6 Utilização dos alinhamentos de texto. 59
4.7 Citações em um texto. 60
4.7 Exemplo de uma citação de um trecho maior. 60
4.8 Como utilizar acentos em sistemas que não suportam acentos. 61
4.8 Exemplo de utilização do comando `input`. Arquivo `livro.tex`. 62
4.9 Exemplo de manipulação direta dos contadores. 63
4.9 Exemplos de formatos de impressão dos contadores. 64

5.1 Inserir Figuras no LaTeX 66
5.2 Posicionamento das Figuras 68
5.3 Rotacionando Figuras 69
5.4 Rotacionando Figuras 69
5.5 Alterando a Largura 71
5.6 Alterando a altura 72
5.7 Alterando a Escala 73
5.8 Cortando a Figura 74
5.9 Opções Conjuntas 75
5.10 Pacote subfigure. 75
5.11 Várias Figuras 75

6.1 Primeiro exemplo simples de tabela. 81
6.2 Exemplo simples de tabela com divisões. 81
6.1 Exemplo de tabela com parágrafo em uma das colunas. 81
6.3 Exemplo de definição múltipla. 83
6.4 Utilização de múltiplas colunas. 83
6.5 Mesclando células ao longo de linhas. 84
6.6 Utilização do ambiente `table`. 84
6.2 Código em `bash` para gerar os dados para a tabela com a representação decimal e binária de um número. 85
6.3 Exemplo de tabela longa. 85
6.4 Exemplo de tabela longa utilizando o `supertabular`. 85

7.1 Listas não numeradas simples. 90
7.2 Alterando o símbolo da lista não numerada simples. 91
7.3 Listas numeradas aninhadas simples. 92

7.4 Listas não numeradas simples com 3 níveis. 92
7.5 Customização de símbolos em listas não numeradas simples com 3 níveis. 93
7.6 Listas Numeradas Simples 94
7.7 Listas Numeradas e Aninhadas 94
7.8 Pacote para Inserir Pacotes para Listas Numeradas em Algarismos Romanos 94
7.9 Exemplo de listas numeradas em algarismos romanos. 95
7.10 Exemplo de listas numeradas em algarismos romanos maiúsculos. 96
7.11 Exemplo de listas aninhadas numeradas em algarismos romanos maiúsculos e minúsculos. 97
7.12 Exemplo de listas com letras minúsculas. 97
7.13 Exemplo de Listas com Letras Maiúsculas 98
7.14 Exemplo de listas aninhadas com letras maiúsculas e minúsculas. 99
7.15 Exemplo de listas aninhadas com descrição. 100
7.16 Exemplo de listas aninhadas com descrição. 100
7.17 Exemplo de listas aninhadas com diferentes tipos. 101
7.18 Exemplo de Lista com 3 Colunas 101

8.1 Pacotes matemáticos. 105
8.2 Ambiente de equações. 105
8.1 Exemplos de utilização das equações. 105
8.3 Frações em LaTeX 106
8.2 Exemplos de utilização de frações e binômios 107
8.3 Exemplos de operadores. 108
8.4 Exemplos do operador para a função cosseno 108
8.4 Exemplos do operador para a função cosseno. 109
8.5 Exemplos de operadores. 109
8.5 Exemplo de definição de um novo operador. 109
8.6 Exemplos de sinais de pontuação 110
8.7 Exemplos de utilização de somatórios, produtórios e integrais 111
8.8 Exemplo simples de matriz. 112
8.9 Exemplos de matrizes. 112
8.10 Exemplo de matriz de uma distribuição conjunta. 113
8.6 Pacote para utilizar o ambiente teorema. 113
8.11 Exemplo de teorema. 113
8.12 Exemplo de demonstração para um teorema. 114
8.13 Exemplo de formatação de caracteres no ambiente matemático.115
8.14 Exemplo de utilização de diacríticos. 116

9.1 Pacote para criar índice remissivo. 122
9.2 Comando para criar o índice remissivo. 122
9.3 Incluindo um índice. 122
9.4 Criando Índice Remissivo. 123

10.1 Uso de bibliografia. 126
10.2 Exemplo de um item de bibliografia 126
10.3 Exemplo de itens no arquivo `.bib`. 127
10.4 Escolhendo o estilo da bibliografia 129

11.1 Inserindo Pacotes Listings 134
11.2 Inserindo Código com Listings 134
11.3 Inserindo Código com Listings 134
11.4 Criando Referências com Listings 135
11.5 Definindo a Linguagem de Programação 135
11.6 Configurações Gerais 135
11.7 Configurações Gerais 136
11.8 Configurações Gerais 137
11.9 Configurações Gerais 137
11.10 Configurações Gerais 138
11.11 Configurações Gerais 139
11.12 Quebra Automática de Linhas 140
11.13 Acentuação e Caracteres Especiais 141

Capítulo 1

Introdução

Sumário

1.1 Palavras iniciais 18

1.1 Palavras iniciais

Antes de iniciar qualquer discussão sobre como criar o seu texto em LaTeX, é necessário que o leitor saiba primeiramente o que é o LaTeX, o que é o TeX e quais as suas funcionalidades. Desta forma o leitor saberá o que pode esperar e quais são as portas que o mundo TeX lhe abrirá.

TeX é um sistema de tipografia que foi projetado e desenvolvido por Donald Knuth. O TeX foi desenvolvido para dois objetivos: 1) permitir a qualquer pessoa produzir os seus próprios livros e documentos, com alta qualidade e mínimo esforço; 2) criar um sistema que obtenha o mesmo resultado visual em todos os computadores. Não há dúvidas sobre a qualidade e reprodutibilidade dos documentos gerados pelo TeX, mas ao longo do seu aprendizado você questionará diversas vezes a sua facilidade de uso. Cada pessoa terá sua própria experiência, mas uma certeza é que hoje está muito mais fácil utilizar o TeX e LaTeX. Existem diversas ferramentas livres disponíveis para ajudar na elaboração do seu texto, além da grande quantidade de documentação, livros, tutoriais e fóruns de discussão. Ao final desta seção voltaremos a este tópico, que será destrinchado ao longo do livro.

Donald Knuth, cientista da computação e professor da Universidade de Stanford, criou o TeX na década de 70 para ser utilizado na edição de seu livro *The Art of Computer Programming*. O TeX se tornou muito popular no meio acadêmico, sobretudo nas áreas das ciências exatas. O sistema TeX já foi utilizado na publicação de milhares de livros, por diversas editoras, como por exemplo a *Addison-Wesley*, *Cambridge University Press*, *Elsevier*, *Oxford University Press* e *Springer*, além disto, diversos jornais científicos aceitam unicamente ou prioritariamente artigos formatados em TeX. O sistema de composição TeX tornou-se popular em várias nichos, e algumas de suas notações passaram a ser utilizadas até mesmo fora do sistema TeX, como é o sistema de composição de fórmulas matemáticas, que é utilizado na Wikipédia. Esta disseminação e universalização do padrão TeX é algo importante que acaba agregando facilidades na composição de um texto utilizando TeX.

Em 1969, foi publicado o primeiro volume do livro *The Art of Computer Programming* de Donald Knuth. Este foi tipografado em prensa de tipos móveis, utilizando letras de metal com um estilo clássico que Knuth apreciava. Quando o segundo volume de seu livro foi publicado, em 1976, a tecnologia de monotipo havia sido substituída pelas técnicas fotográficas. Knuth não gostou da prova feita e decidiu desenvolver um sistema de tipografia digital de alta qualidade. A primeira versão do TeX foi disponibilizada em 1978, mas a linguagem utilizada pelo TeX só foi congelada em 1989.

Após ser congelado, a partir da versão 3, só são lançadas correções de

bugs. Knuth mantém registro detalhado de todos os bugs encontrados e correções desde 1982. Ele passou a oferecer recompensa financeira para aqueles que encontrassem bugs no TeX. A recompensa inicial começou em U$1,28 e dobrava a cada bug encontrado, até ser congelada no valor de U$327,68. Knuth não levou muito prejuízo, uma vez que poucos bugs foram encontrados e a maioria das pessoas preferia guardar os cheques como prova de que encontraram um bug a sacar o valor. Os números das versões subsequentes do TeX aproximam assintoticamente ao valor de π. A versão atual (atualizada em Janeiro de 2014) é de número 3.14159265.

Os texto feitos em TeX são obviamente digitados em um computador. Sabemos que o teclado do computador possui um número finito de símbolos que podem ser utilizados. Um texto genérico pode contar vários outros símbolos, além daqueles disponíveis no teclado, como letras gregas (α, β, γ, ...), letras do alfabeto com acento gráfico (á, ẽ, ï, ò, ŷ, ç, ...), símbolos matemáticos ($\neq$, $\in$, $\forall$, $\rightarrow$, ...) e muitos outros. Para gerar símbolos, criar tabelas e figuras, além de controlar a formatação do texto, o sistema de composição do TeX utiliza comandos que usualmente são denotados por uma sequências de caracteres iniciada por uma barra inversa (ou barra invertida, contrabarra; "\") e são agrupados por chaves ("{.}"). O sistema TeX é baseado em alguns comandos básicos, chamados de *primitivas*, que são raramente utilizados diretamente pelo usuário. As primitivas são os blocos (átomos) utilizados para a construção de outros comandos mais complexos. Logo, as primitivas não podem ser divididas/expressas em termos de outros comandos. A partir das primitivas cria-se comandos cada vez mais complexos que podem também ser elaborados a partir de comandos mais simples, mesmo que não sejam primitivas em si. Existem aproximadamente 900 comandos em TeX, sendo que em torno de 300 deles são primitivas [3]. Muitos dos comandos que utilizamos são, em verdade, uma composição de diversos outros comandos. Usualmente as funcionalidades manipuladas diretamente pelos usuários são fornecidas por arquivos de formato e pacotes, que são uma coleção de macros para TeX. O formato mais utilizado é o LaTeX, criado por Leslie Lamport, que incorpora estilos para livros, artigos, slides, etc., adiciona suporte para referências, numeração automática de sessões e equações, etc.

O código fonte do TeX está em "domínio público", mas estritamente está protegido por *copyright*, então tecnicamente não é domínio público, mas sim software livre/código-aberto. Knuth encoraja as modificações e experimentações feitas a partir do TeX, mas qualquer versão derivada não pode ser chamada de TeX.

Para entender um pouco como um TeX funciona, veremos como seria em TeX um simples código de *Hello Word*:

```
Hello World!
\bye          % marca o final de arquivo
```

E um simples texto com uma fórmula seria codificado como:

```
Equação de uma parábola: $y = ax^2 + bx + c$
\bye
```

que seria visualizado da seguinte forma:

Equação de uma parábola: $y = ax^2 + bx + c$

Para utilizar o TeX, você pode digitar os comandos diretamente no interpretador de comandos ou gravar a sequência de comandos em um arquivo. A segunda alternativa é o modo usual de utilização do TeX, uma vez que possibilita que o código seja reutilizado, editado, corrigido, etc. Você pode editar um arquivo com comandos TeX até mesmo em um bloco de notas. Os editores para TeX, de forma geral, não são WYSIWYG[1]. Existem editores para diversas plataformas: Linux, Windows, MacOS, Unix, etc., mas qualquer editor de texto pode ser utilizado para escrever os códigos TeX e salvar em um arquivo de puro texto para ser compilado.

O LaTeX é um sistema de preparação de documentos e uma linguagem de marcação criado para o sistema de tipografia TeX. O propósito do LaTeX é agregar diversas funcionalidades ao TeX de forma que os autores possam focar no que estão escrevendo, sem a distração causada pela apresentação visual do texto. Com a utilização do LaTeX o autor é capaz de dissociar conteúdo de forma, podendo se concentrar em cada um deles separadamente. Ao escrever um documento utilizando LaTeX, o autor define apenas as estruturas básicas do texto, tais como capítulos, sessões, tabelas, figuras, etc., deixando a apresentação do texto a cargo do LaTeX, e podendo fazer suas customizações posteriormente.

Um exemplo simples de um texto feito em LaTeX segue abaixo, e o resultado visual obtido pode ser visto na Listagem 1.1.

Listagem 1.1: Exemplo de Documento em Latex

```
\usepackage{amsmath}
\title{Exemplo}
\date{}
\begin{document}
```

[1]WYSIWYG é a sigla em inglês para "What You See Is What You Get". O termos costuma ser utilizado para classificar ferramentas de edição que permitem visualização do resultado simultaneamente ao desenvolvimento.

```
\maketitle
Simples exemplo de documento feito em \LaTeX{}.
% comentários são precedidos pelo simbolo de
   porcento
\begin{equation}
y = ax^2 + bx + c
\end{equation}
```

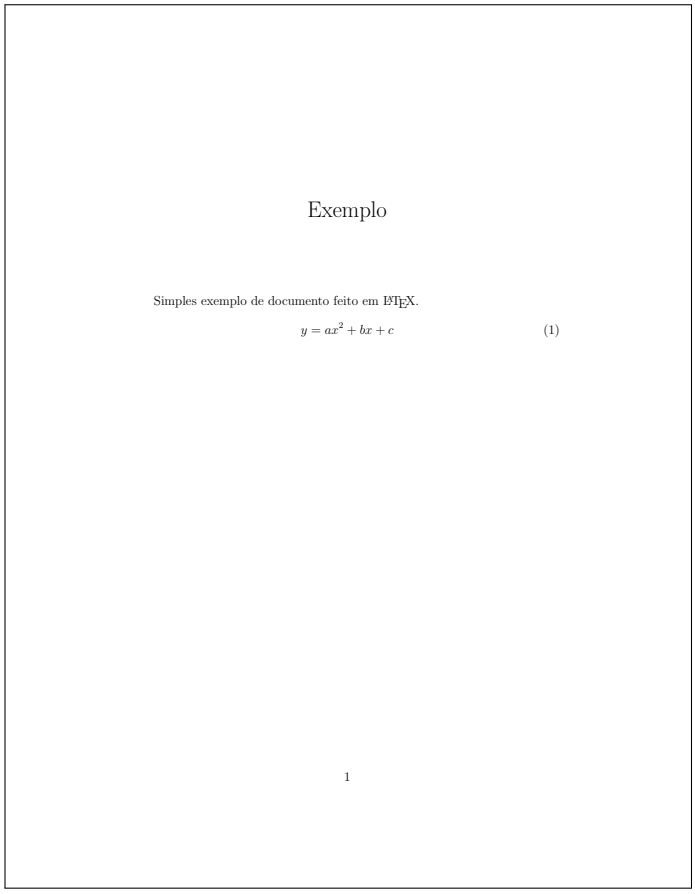


Figura 1.1: Exemplo de um Documento em LaTeX

Inicialmente escrever um texto utilizando LaTeX pode parecer complicado, mas depois de um tempo de aprendizado as vantagens compensam o esforço e produzir textos deixa de ser uma tarefa árdua, sobretudo ao produzir textos grandes e complexos. Existem várias vantagens em se utilizar o LaTeX na produção de textos, vamos tentar listar aqui algumas. Cada usuá-

rio, com sua experiência na utilização deste sistema de tipografia, poderá julgar quais são os benefícios para si.

Uma primeira vantagem na utilização do LaTeX é a sua portabilidade. Como o sistema está implementado em diferentes plataformas (Mac OS, Solaris, Linux, BSDs, OpenSolaris, Microsoft Windows, RISC OS e Amiga OS) é possível levar o seu documento de uma plataforma para outra sem a necessidade de conversão. Os documentos criados em um sistema podem ser compilados em outro sistema e o resultado gerado será o mesmo. Mesmo que um novo sistema surja, o sistema de tipografia TeX poderá ser implementado neste sistema, e como o padrão é fixo e imutável, um documento produzido hoje poderá ser apresentado exatamente com o mesmo visual em qualquer sistema e em qualquer instante no tempo. O padrão utilizado é aberto e novas ferramentas podem ser criadas para facilitar a produção de textos, criando facilidades e buscando atingir o visual desejado pelo usuário.

A codificação dos arquivos fontes é feita em arquivos de puro texto, desta forma é fácil utilizar sistemas de controle de versão (como Git ou CVS/SVN) para controlar as alterações feitas no documento, verificar o histórico de alterações e comparar versões. Se por ventura o arquivo fonte for corrompido, talvez ainda seja possível recuperar parte dele, mesmo sem um backup ou sistema de controle de versão. Isto raramente é possível de ser feito em arquivos do Microsoft Word, quando uma parte considerável do arquivo é danificada, acarretando a perda de todo trabalho.

Os documentos preparados em LaTeX são visivelmente mais elegantes e o autor consegue produzir o que quiser, pois possui controle sobre o resultado que será gerado. A codificação de fórmulas matemáticas é muito mais eficiente e simples. Criar referências e fazer citações no texto também é muito prático. Preparar um texto muito extenso com o LaTeX é uma tarefa mais fácil do que utilizar os processadores de texto convencionais, além disso a estabilidade é um critério muito importante, ao redigir textos longos, utilizando muitas tabelas e figuras. Afinal, não queremos um editor emperrando o computador ao lidar com trabalhos extensos.

Com a utilização do LaTeX não ocorrerão erros causados pela instabilidade do software. A Figura 1.2 ilustra a relação entre complexidade e tamanho de um documento com o esforço e tempo gasto para produzir, comparando a utilização do LaTeX com o Word. Embora o Word seja prático para criar documentos breves e simples, ele torna-se muito complexo e até mesmo inutilizável para criar documentos muito extensos e complexos. O LaTeX requer um maior esforço inicial e tempo de aprendizado, mesmo para criar documentos pequenos e simples, mas após este aprendizado, a tarefa de criar documentos, mesmo documentos longos e complexos torna-se mais

fácil e direta.

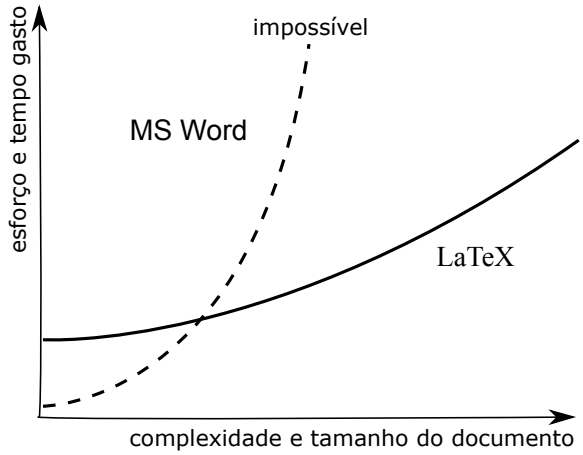


Figura 1.2: Comparativo entre LaTeX e outros Editores (figura adaptada de [10]).

Além destas vantagens apresentadas, o LaTeX é completamente grátis. Você pode utilizar para redigir seus textos ou para publicar textos sem nenhum custo. Você pode instalar em quantas máquinas quiser e utilizar por um tempo indeterminado, sem a necessidade de adquirir licenças ou autorizações.

Nesse livro não buscaremos exaurir todas as possibilidades que o LaTeX nos fornece, em verdade iremos cobrir apenas os tópicos que julgamos essenciais para começar a utilizar o LaTeX e algumas possibilidades que, embora não sejam tão frequentes, iremos abordá-las para que o leitor tenha uma fonte para consulta posterior. A documentação disponível do TeX e LaTeX é bem ampla e facilmente obtida na Internet. Muitas vezes, para resolver dúvidas sobre como fazer algo em documento basta realizar uma busca online. Além de encontrar várias respostas, existe uma grande comunidade de usuários dispostos a ajudar e resolver dúvidas de outras pessoas, através de fóruns e comunidades.

Uma das maiores frustrações que iniciantes, e até mesmo usuários experientes, podem encontrar ao utilizarem LaTeX é a falta de flexibilidade para atingir um design e *layout* específico. De forma geral, LaTeX faz automaticamente a formatação para você, mas isto não impede que você crie sua formatação personalizada. Você pode criar suas próprias macros ou utilizar aquelas criadas por outros usuários. Várias pessoas provavelmente já passaram pelos mesmos problemas e é bem provável que existam macros criadas

por outros usuários e publicadas na forma de pacotes para atingir uma formatação ou funcionalidade desejada. Um bom lugar para encontrar pacotes para TeX é no site do CTAN[2].

Hoje o TeX possui uma família de diversas ferramentas que derivaram dele, como o LaTeX mencionado anteriormente, o pdfTeX, XeTeX, LuaTeX, etc. O TeX original produz um documento formatado do padrão DVI (*DeVice Independent format*) que pode ser posteriormente convertido em PostScript para impressão. Em 1993 surgiu o formato PDF com inovações como o suporte a hyperlinks, secção de metadados e suporte a novos formatos de imagem (por exemplo, `.png`, `.jpg` e `.pdf`). O pdfTeX foi criado por Hàn Thê Thánh para permitir a criação direta de um arquivo PDF, usufruindo assim dessas novas funcionalidades. O XeTeX permitiu a utilização de uma maior extensão de diferentes caracteres. Já o LuaTeX utiliza a linguagem de scripts Lua para fornecer uma forma mais simples de programação.

No próximo capítulo iremos explicar os procedimentos de instalação no Windows e Linux. Em seguida iremos iniciar propriamente as explicações de como criar documentos em LaTeX.

[2]CTAN (*The Comprehensive TeX Archive Network*) acessível no sítio `https://www.ctan.org/`.

Capítulo 2

Instalação das Ferramentas

Sumário

2.1 Visão Geral da Instalação 26

2.2 Instalação do Compilador de LaTeX 26

2.2.1 Linux 26

2.2.2 Instalação do Latex do Windows 27

2.2.3 Instalação do LaTeX no Mac OS 28

2.3 Instalando os Editores 29

2.3.1 Instalação do TexMaker 29

2.4 Gerenciadores de Bibliografia 29

2.1 Visão Geral da Instalação

Para produzir documentos utilizando o LaTeX é necessário instalar ao menos o compilador, mas se quiser ter em mãos algumas facilidades, recomendamos instalar 3 softwares distintos:

1. o compilador de TeX (obrigatório);
2. um editor para LaTeX ou usar qualquer editor de textos já instalado em seu sistema operacional;
3. um software gerenciador de referências bibliográficas também pode ser útil.

Lembre-se que estes softwares dependem do Sistema Operacional (SO) que está instalado em seu computador. Existem softwares específicos para cada plataforma, combinação entre Sistema Operacional e Hardware, existente: Windows , Mac OS X e Linux .

Caso não queira instalar nenhum software você pode utilizar o seu navegador e produzir o seu documento gratuitamente no site Overleaf (`https://www.overleaf.com`).

2.2 Instalação do Compilador de LaTeX

Nesta seção serão detalhados os procedimentos necessários para instalar o compilador LaTeX para todos os Sistemas Operacionais. A Seção 2.2.1 demonstrará os passos para instalação no Linux, na Seção 2.2.3 no Mac OS X e na Seção 2.2.2 no Windows.

2.2.1 Linux

Para instalar o Ambiente LaTeX no Sistema Operacional Linux abra o `Shell` ou gerenciador de programas e instale os seguintes pacotes:

- texlive
- texlive-latex-extra
- texlive-lang-portuguese
- texlive-math-extra

Para instalar via `Shell`, nas distribuições Linux da família Debian, basta utilizar o código na Listagem 2.1.

Listagem 2.1: Instalação do Compilador Latex em distribuições Linux da família Debian.

```
# instalação básica
sudo apt-get install texlive
# extras recomendáveis
sudo apt-get install texlive-latex-extra
sudo apt-get install texlive-lang-portuguese
sudo apt-get install texlive-math-extra
```

Outra alternativa é instalar todos os pacotes do LaTeX como na Listagem 2.2

Listagem 2.2: Instalação Completa do LaTeX

```
sudo apt-get install texlive-full
```

Em outras distribuições Linux, basta utilizar o gerenciador de pacotes de cada uma. O ArchLinux utiliza o *pacman*, o Fedora/Redhad utilizam o *dnf*, Gentoo utiliza o *emerge*, e assim por diante. Consulte a documentação da sua distribuição para verificar como instalar um pacote.

2.2.2 Instalação do Latex do Windows

O TeX Live descrito na seção anterior também está disponível para instalação no Sistema Operacional `Windows`. No site oficial `http://www.tug.org/texlive/` você encontrará o instalador para Windows: `install-tl-windows.exe`. Outra opção muito comum é utilizar outra distribuição TeX/LaTeX para Microsoft Windows, o `Miktex` , disponível no seguinte sítio `https://miktex.org/howto/install-miktex` ou em seu repositório no GitHub (`https://github.com/MiKTeX/miktex`). Basta baixar o instalador (veja a Figura 2.1) e executá-lo. O Miktex também está disponível para outros Sistemas Operacionais como Linux e Mac OS X.

Para decidir qual software baixar é necessário verificar se seu processador é de 32 ou 64 bits. Execute o instalador e escolha a opção para a instalação completa para evitar problemas de baixar pacotes adicionais.

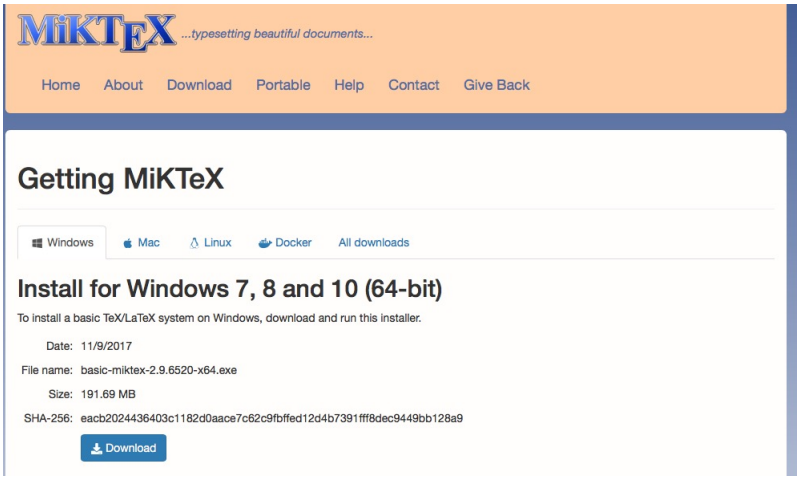


Figura 2.1: Site do Miktex.

2.2.3 Instalação do LaTeX no Mac OS

Para instalar no Sistema Operacional `OSX` basta acessar o site `MacTex` [7] e baixar o instalador. O procedimento de instalação é semelhante a qualquer instalação de outro software. A Figura 2.2 apresenta o site para baixar o instalador.

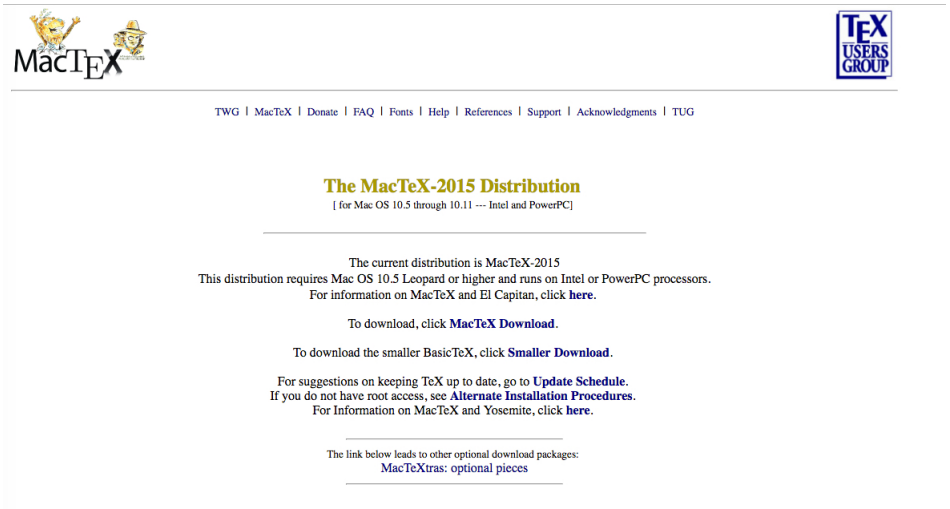


Figura 2.2: Site do MacTex.

2.3 Instalando os Editores

Existem diversos editores de LaTeX como o Texmaker [11], TexShop [6], TeXstudio [12] e o TeXworks [13]. Existem outras versões que são online, como o Overleaf [8], e outras nativas para alguns sistemas operacionais como Kile [2].

A escolha do software de edição é pessoal e por isto não vamos focar no uso de um ou de outro editor. Pessoalmente, os autores deste livro costumam utilizar o Texmaker, o Vim e o Overleaf. O Vim é o mais leve de todos, possui muitos recursos, mas é difícil para um iniciante. O Texmaker é uma boa relação de compromisso, é simples e leve, mas fornece vários recursos facilmente utilizados. Existem diversos outros editores com recursos bem mais avançados, porém não são leves. O Overleaf depende de acesso a Internet, mas é excelente para redigir um texto em colaboração com outros pessoas e realiza controle de versão automaticamente utilizando Git. Uma boa sugestão de leitura é o sítio *10 Best LaTeX Editors You Should Use*[1] faz uma descrição detalhada de todos os editores.

2.3.1 Instalação do TexMaker

O editor Texmaker pode ser obtido através do sítio `http://www.xm1math.net/texmaker/download.html`. Ele dispõe de versões para os seguintes SOs: Windows, Mac OS X, Linux e OS/2. Para o Linux pode ser instalado via Shell como ilustrado na Listagem 2.3.

Listagem 2.3: Instalação do Texmaker no Linux.

```
sudo apt-get install texmaker
```

2.4 Gerenciadores de Bibliografia

Estes softwares funcionam como um banco de dados especializado para auxiliar o gerenciamento de referências bibliográficas. Eles auxiliam na citação, armazenamento e busca de referências bibliográficas. como o de armazenamento e de busca de referências.

Existem diversos softwares gerenciadores de bibliografia disponíveis para download. Se quiser saber detalhes sobre grande parte dos softwares existentes verifique o Estudo Comparativo disponível em [14].

Neste livro optamos pelo uso do Mendeley (`https://www.mendeley.com/`), pois é multiplataforma e por possuir um site que armazena todas as suas citações. Através desta funcionalidade o usuário poderá baixar suas referências para qualquer dispositivo.

Capítulo 3

Primeiro Documento

Sumário

3.1 Sintaxe **32**
3.2 Comentários **33**
3.3 Codificação e suporte a diferentes línguas **34**
3.3.1 Codificação de caracteres 35
3.4 Codificação de fonte **40**
3.5 Anatomia do documento **41**
3.6 Pacotes **44**
3.7 Divisão do documento **45**
3.8 Ambientes **46**
3.9 Tipografia das seções **46**
3.10 Um exemplo de documento mais completo . . . **47**

3.1 Sintaxe

O LaTeX normaliza os `espaços`, de forma que múltiplos espaços, `tab` e quebra de linha serão compreendidos como um único espaço simples. Espaços no início de uma linha serão ignorados, uma única quebra de linha será entendida como um espaço simples, já duas ou mais quebras de linhas serão interpretadas como o fim de um parágrafo. O exemplo abaixo ilustra o resultado da utilização de diferentes espaços.

Listagem 3.1: Trabalhando com espacos.

```
Independente do numero de     espacos entre   as
palavras    ou se ha quebra de linha entre elas
todas               serao tratados como um
espaco simples.

E necessário deixar uma linha em branco para fazer
o fim de um paragrafo.
```

Existe alguns poucos caracteres reservados em LaTeX, são eles: #, $, %, ^, &, _, {, }, ~ e \. Este caracteres possuem significado especial e portanto não serão tratados como meros caracteres. Para exibir estes caracteres em texto é necessário utilizar um comando específico para cada um deles. Os comandos são, respectivamente: \#, \$, \%, \^, \&, _, \{, \}, \~ e \\{}. A barra ao contrário (\) não poderá ser gerada utilizando-se duas em sequência, pois a sequência \\ está reservada para quebra de linha. Os comandos \~ e \^ produzem os acentos que serão exibidos sobre a próxima letra. Por exemplo, para escrever um ã devemos utilizar o comando \~{a}.

Diversos ambientes estão definidos no LaTeX e ainda é possível definir outros. Para delimitar um ambiente deve-se utilizar os comandos `begin` e `end` e utilizar o nome do ambiente entre chaves. Por exemplos, formulas matemáticas são definidas dentro do ambiente `equation`. Veja o exemplo.

Listagem 3.2: Ambiente `equation`.

```
\begin{equation}
    f(x) = \int_{-\infty}^x g(\tau) d\tau
\end{equation}
```

Os comandos em LaTeX sempre iniciam-se com \, em seguida segue o nome do comando, propriamente dito. É importante enfatizar que o LaTeX é sensível às diferenças de maiúsculas e minúsculas. O nome de um comando termina ao encontrarmos um espaço em branco ou qualquer caractere não alfabético. Os comandos podem ainda possuir argumentos, que serão fornecidos entre chaves `{}`; e/ou parâmetros opcionais que serão fornecidos entre colchetes `[]`. Desta forma teremos a seguinte sintaxe para evocar um comando:

Listagem 3.3: Sintaxe de comandos em LaTeX.

```
1 \nomedocomando[opção1,opção2,...]{argumento1}{argumento2}...
```

Por exemplo, o comando para escrever algo em negrito é `\textbf`, este requer como parâmetro apenas o texto que aparecerá em negrito. Apresentamos também um exemplo de comando que utiliza-se de dois parâmetros, o comando `\rule` recebe como parâmetro o comprimento do traço e a espessura da linha. Veja o exemplo de utilização.

Listagem 3.4: Utilização do comando `\textbf` para escrever em negrito.

```
1 Este texto foi escrito em \textbf{negrito}. \rule{1cm}{0.4pt}
```

3.2 Comentários

Em um arquivo `.tex` tudo aquilo que vier após o caractere `%` será considerado um comentário, ou seja, um trecho que será suprimido do documento. Este recurso é muito utilizado na elaboração de documentos em LaTeX. Muitas vezes queremos adicionar alguma informação que não será exibida no documento final, algum comentário que será útil para o autor que está elaborando o documento. Às vezes queremos suprimir um trecho de texto que eventualmente poderá ser novamente incorporado. Podemos também utilizar comentários para explicitar o que alguma sequência de comandos do LaTeX faz, para ficar mais fácil relembrarmos da próxima vez que nos depararmos com aquele trecho no documento. Enfim, o comentário pode servir a qualquer fim que você julgar necessário. Tudo que você precisa lembrar é que o LaTeX irá ignorar tudo aquilo que vier após o caractere `%`. Veja um exemplo de utilização na Listagem 3.5.

Listagem 3.5: Comentários.

```
Este é nosso documento.
% Este comentário não será exibido.
% Utilizamos o comando \LaTeX{} para escrever
%    LaTeX de forma estilizada.
O documento foi feito em \LaTeX{}.
```

O LaTeX não possui uma forma própria de fazer comentários em bloco. Caso queira fazer um comentário com múltiplas linhas, deverá colocar `%` no início de cada uma (alguns editores possuem atalhos para fazer isso), ou usar algum artifício. Podemos, por exemplo, usar o pacote `verbatim` que disponibiliza o ambiente `comment`. Veja o exemplo da Listagem 3.6.

Listagem 3.6: Comentário em bloco utilizando `comment`.

```
\usepackage{verbatim}
...
\begin{comment}
Este e um comentario.
Utilizamos varias linhas.
O conteudo dele nao sera exibido.
\end{comment}
```

3.3 Codificação e suporte a diferentes línguas

Como já dito anteriormente, os documentos em LaTeX são armazenados em um arquivo texto (`.tex`) e desta forma é importante saber qual é o tipo de codificação que está sendo utilizado para armazená-lo, ou seja, qual o padrão utilizado para codificar os caracteres em palavras binárias. Os computadores modernos permitem que caracteres especiais presentes nas mais diferentes línguas sejam inseridos diretamente através do teclado (físico ou virtual) e serão codificados de acordo com a codificação utilizada pelo seu sistema operacional. Para escolher o tipo de codificação que será utilizada no documento LaTeX deve-se usar o comando:

Listagem 3.7: Especificando a codificação do seu código `.tex`.

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

onde você deverá escolher o parâmetro `codificação` de acordo com o tipo de codificação. Sistemas Unix usualmente utilizam `utf8`, enquanto Windows mais antigos usam `latin1`. É recomendável usar sempre `utf8` que vem se tornando um padrão em todos sistemas operacionais.

Para ativar o suporte às mais diferentes línguas, deve-se ativar o pacote `babel` e use como parâmetro a língua que deseja utilizar. Basta inserir no preâmbulo do documento a linha:

Listagem 3.8: Especificando a língua do documento.

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

onde o parâmetro `lingua` deverá ser o idioma desejado. Usualmente usamos:

Listagem 3.9: Especificando codificação e língua para um documento.

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

3.3.1 Codificação de caracteres

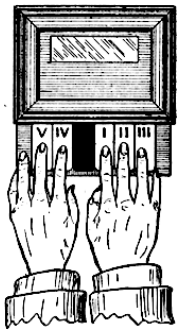
A título de curiosidade, nesta seção apresentaremos uma breve descrição do histórico e dos padrões de codificação de caracteres utilizados em computadores. É interessante conhecer um pouco sobre este tópico para saber exatamente com o que estamos lidando e entender possíveis erros que podem surgir quando especificamos uma codificação errada, ou quando reaproveitamos textos já escritos, mas que eventualmente podem estar utilizando uma codificação diferente. Para uma visão mais detalhada e abrangente do histórico das codificações, recomendamos a leitura do texto *The Evolution of Character Codes, 1874-1968* de autoria de Eric Fischer. O texto está acessível através do seguinte sítio: `https://github.com/ericfischer/ascii`.

O código Morse pode ser compreendido como um precursor dos códigos digitais. Samuel Morse havia sido comissionado em 1825 para pintar um retrato de Lafayette, em uma visita a Washington, DC, veja a Figura 3.1. Enquanto pintava, ele recebeu uma mensagem avisando que sua esposa estava muito doente. Morse partiu imediatamente para sua casa em New Haven. Quando chegou sua esposa já havia sido enterrada. Ele decidiu

Figura 3.1: Retrato do Marquês de Lafayette feito por Samuel Morse em 1825 (Wikimedia: domínio público).

então se dedicar a explorar formas de comunicações a longa distância que fossem mais rápidas. A primeira versão do código, desenvolvida por Morse durante uma viagem transatlântica em 1832, era mais complexa do que a versão estabelecida em 1843. Mais tarde, Morse abandonou sua versão em favor dos conhecidos pontos e traços desenvolvidos em conjunto com Alfred Vail. Morse recebeu a patente do seu telégrafo com um único fio em 1847, sobrepujando o telégrafo de múltiplos fios proposto por Cooke e Wheatstone, que havia sido patenteado em 1837. Para que a comunicação fosse mais eficiente Morse e Vail se preocuparam em utilizar códigos curtos para representar letras frequentes, deixando códigos longos para as letras menos frequentes. Na época não havia nenhum estudo sobre a frequência de ocorrência dos caracteres na língua. O que fizeram então foi visitar gráficas e verificar quantos tipos gráficos eles possuíam para cada letra, o que seria uma boa aproximação para a frequência de uso das letras na língua.

Na França, Emile Baudout projetou seus sistema para o telégrafo em 1874. Seu código foi baseado em código anterior desenvolvidos por Carl Friedrich Gauss e Wilhelm Weber em 1834. Todos os símbolos possuem o mesmo comprimento: cinco. O projeto utilizava um conjunto de fios funcionando de forma síncrona em um sistema de multiplexação, onde o operador



(a) Teclado utilizado no código Baudout.

	1	2	3	4	5
A	+	−	−	−	−
B	−	−	+	+	−
C	+	−	+	+	−
D	+	+	+	+	−
E	−	+	−	−	−
F	+	+	−	−	−
G	−	+	+	+	−
H	+	+	−	+	−
I	−	+	+	−	−
J	+	−	−	+	−
K	+	−	−	+	+
L	+	+	−	+	+
M	−	+	−	+	+
N	−	+	+	+	+
O	+	+	+	−	−
P	+	+	+	+	+
Q	+	−	+	+	+
R	−	−	+	+	+
S	−	−	+	−	+
T	+	−	+	−	+
U	+	−	+	−	−
V	+	+	+	−	+
W	−	+	+	−	+
X	−	+	−	−	+
Y	−	−	+	−	−
Z	+	+	−	−	+
↵	+	−	−	−	+
⇧	−	−	−	+	+
	−	−	−	+	−
	−	−	−	−	+
	−	−	−	−	−

(b) Codificação dos caracteres segundo o código Baudout.

Figura 3.2: Código Baudout [9].

humano era responsável por realizar a divisão temporal e assim a sincronização. Os códigos eram gerados por um aparelho com cinco teclas (similar às teclas de um piano), sendo operado com duas mãos (dois dedos da mãos esquerda e três da mão direita, veja a Figura 3.2b). Quando ordenados em alfabeticamente, as vogais e as consoantes, formam um código de Gray. O código Baudout foi projetado para minimizar os movimentos da mão e dedos, reduzindo assim a fadiga.

Mais tarde, o código de Baudout foi modificado por Donald Murray (1901) para ser utilizado em um aparelho com teclado QWERTY. A mensagem é gravada em uma fita através de perfurações e transmitida a partir desta fita perfurada. Deixou assim de existir a conexão direta entre a mão do operador e a informação transmitida, não sendo mais necessário preocuparse com a fadiga. O objetivo passou então a ser simplificar o equipamento e minimizar seu desgaste, para tanto as combinações com menos buracos foram utilizadas para designar caracteres mais frequentes (no inglês temos a seguinte ordem: e,t,a,o,i,n,s,h,r,d,l,c,u,m,w,f,g,y,p,b,v,k,j,x,q,z). O código Murray também introduziu os caracteres de controle CR (carriage return) e LF (line feed). A Figura 3.3 apresenta o código Baudot-Murray utilizado para o inglês.

O código Murray foi adotado pelo Western Union com algumas modifi-

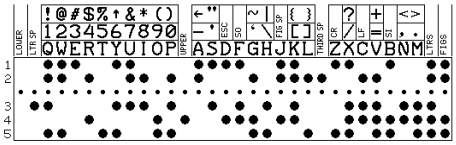


Figura 3.3: Código Baudot-Murray adaptado (ITA2). (`http://www.quadibloc.com/crypto/images/improved.gif`)

cações, sendo utilizado até os anos 50. Em 1924 o *International Telegraph and Telephone Consultative Committee* criou o ITA2 (*international telegraph alphabet* n. 2), baseado no código da Western Union. ITA2, também chamado de US TTY (*American Teletypewriter code*) foi a base para codificação em 5 bits dos Teletipos até o surgimento do código de 7 bits, ASCII em 1963.

O ASCII foi desenvolvido pelo Comitê X3 da ASA (American Standards Association), da qual faziam parte IBM (embora só passou a adotar o ASCII na década de 80), AT&T e sua subsidiária Teletype Corporation. Os caracteres estão organizados de forma que os caracteres alfabéticos, numéricos, matemáticos e de controle podem ser isolados através de uma simples máscara binária. O caractere A fica na posição 41_{hex} para ser compatível com o padrão britânico. Os dígitos de 0 a 9 começam com 011 e a sequência binária seguinte corresponde ao valor binários de cada um deles, facilitando assim a conversão decimal-binário. Os caracteres !"#$%&() foram adicionados à 2 coluna de forma a melhor se adequarem à posição que ocupavam nos teclados das máquinas de escrever, de forma que a tecla *shift* corresponderia à uma simples mudança de um bit, assim facilitando a compatibilidade com as máquinas de escrever. Foi cogitado utilizar um código com 8 bits, de forma que dois padrões de 4 bits codificariam 2 dígitos. Isto iria requerer que fosse enviado sempre 8 bits. Para minimizar custos, adotou-se 7 bits. Como as fitas perfuradas podiam armazenar 8 bits em cada posição, seria ainda possível utilizar um bit de paridade se desejado.

Com a disseminação de computadores em todo o mundo surgiu a necessidade de representar caracteres das mais variadas língua. O código ASCII já não era suficiente. Foi necessário criar novos tipos de codificação. A popularização do IBM System/360 e microprocessadores como o Intel 8008, 8080 e 8086 acarretou na padronização do byte como uma unidade de 8 bits. Endereçamento e armazenamento passaram a ser feitos em 8 bits, possibilitando assim a extensão do ASCII utilizando o bit extra. Seria assim possível representar outros 128 caracteres, além daqueles do ASCII original, sem custo adicional. Várias extensões do ASCII surgiram nesta época. A ISO acabou

USASCII code chart

b_7 b_6 b_5 →					0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
Bits b_4	b_3	b_2	b_1	Row ↓ / Column →	0	1	2	3	4	5	6	7
0	0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	2	STX	DC2	"	2	B	R	b	r
0	0	1	1	3	ETX	DC3	#	3	C	S	c	s
0	1	0	0	4	EOT	DC4	$	4	D	T	d	t
0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
1	0	0	0	8	BS	CAN	(	8	H	X	h	x
1	0	0	1	9	HT	EM	)	9	I	Y	i	y
1	0	1	0	10	LF	SUB	*	:	J	Z	j	z
1	0	1	1	11	VT	ESC	+	;	K	[	k	{
1	1	0	0	12	FF	FS	,	<	L	\	l	|
1	1	0	1	13	CR	GS	–	=	M	]	m	}
1	1	1	0	14	SO	RS	.	>	N	^	n	~
1	1	1	1	15	SI	US	/	?	O	—	o	DEL

Figura 3.4: Tabela ASCII contida no manual TermiNet 300 da GE (`http://archive.computerhistory.org/resources/text/GE/GE.TermiNet300.1971.102646207.pdf`)

lançando o padrão ISO 8859 descrevendo sua própria extensão do ASCII com 8 bits. O mais popular foi o ISO 8859-1, também conhecido como ISO Latin 1, abrangendo os caracteres das línguas da Europa Ocidental e América Latina. O Windows-1252 ou CP-1252 foi criado pela Microsoft utilizando as regiões não utilizadas pelo ISO 8859-1. Criou-se assim um código compatível com o ISO 8859-1, porém com caracteres extras.

Na década de 1980 começaram os esforços para superar as limitações impostas pelos sistemas de codificação da época. Formou-se então um consórcio para criar uma padronização para realizar uma codificação consistente, capaz de representar qualquer texto expresso em grande parte das línguas do mundo. Surge então a padronização do Unicode com sua primeira versão em 1991. O Unicode pode ser implementado por diferentes codificações de caracteres. O padrão Unicode define o UTF-8, UTF-16, e UTF-32. Atualmente o mais utilizado é o UTF-8. Este é um formato de codificação multi-byte e compatível (compatibilidade reversa) com ASCII, pois trata-se de uma extensão do mesmo. Os códigos em UTF-8 utilizam de 1 a 6 bytes, sendo capaz de representar até 2.216.757.376 pontos de codificação do Unicode. Os símbolos que são representados pelo ASCII são representados em UTF-8 por um único octeto com mesmo valor binário que o ASCII precedido por um 0. Os pontos de código mais usuais utilizam menos bytes que aqueles

menos comuns, diminuindo assim o comprimento esperado do código. 1920 caracteres utilizam 2 bytes para representar o restante do alfabeto latino (romano), grego, cirílico, copta, armênio, hebreu, arábico, siríaco, thaana e n'ko. Para as demais línguas são utilizados 3 bytes ou 4 bytes para representar caracteres como símbolos matemáticos e emojis. O primeiro byte determina o número de bytes na sequência (veja a Tabela 3.1). UTF-8 foi apresentado em uma conferência em 1993. Em 2003 foi registrado pela RFC 3629 e, a partir de 2008, tornou-se o padrão mais utilizado na internet.

Tabela 3.1: Estrutura de codificação do UTF-8.

número de bytes	bits por ponto de codificação	primeiro ponto de codificação	último ponto de codificação	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6
1	7	U+0000	U+007F	0xxxxxxx					
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx				
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx			
4	21	U+10000	U+1FFFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx		
5	26	U+200000	U+3FFFFFF	111110xx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	
6	31	U+4000000	U+7FFFFFFF	1111110x	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx	10xxxxxx

Outro fator que pode variar da representação binárias dos dados é a extremidade (*endianness*). A extremidade refere-se a ordem utilizada para armazenar/ler os bytes ou bits de dados. Na ordenação *big-endian*, o byte mais significativo está no endereço mais baixo, enquanto na ordenação *little endian*, o byte menos significativo está no endereço mais baixo. Quando a numeração dos bits começa em zero para o bit menos significativo, o esquema de numeração é chamado LSB 0. Quando a numeração dos bits inicia-se pelo mais significante, chama-se MSB 0.

Com todas estas possíveis variações, devemos nos atentar para utilizar uma codificação de forma consistente e definir corretamente a codificação de entrada do nosso arquivo `.tex`. Em princípio não é possível determinar automaticamente de forma segura a codificação de um arquivo. Programas que verificam a codificação de um arquivo buscam fazer uma adivinhação educada, ou seja, buscam qual é a codificação mais provável para um determinado arquivo que se deseje testar.

3.4 Codificação de fonte

A codificação de fonte faz o mapeamento entre códigos de caracteres e glifos. Você poderá escolher explicitamente a codificação que seja ou, caso não especifique, será utilizada a codificação `OT1`, padrão do LaTeX. Essa codificação foi criada por Donald Knuth no TeX. Ela é um código de 7 bits capaz de representar 128 glifos, não incluindo assim letras com acento. Isto não signi-

fica que não seja possível fazer letras com acento utilizando esta codificação, mas isto significa que para fazê-las será necessário realizar uma composição. Desta forma os glifos com acento não serão tratados como um único símbolo, o que pode acarretar incapacidade de realizar hifenização e também erros ao copiar e colar os textos gerados de saída (DVI/PS/PDF). Para evitar essas falhas é recomendável utilizar uma codificação de fonte adequada à língua em que o texto esteja sendo escrito, ou seja, uma codificação com suportes aos símbolos presentes no alfabeto desta língua.

Para ter suporte aos símbolos presentes em grande parte das línguas da Europa Ocidental (Português, inclusive) e algumas da Europa Oriental, foi desenvolvida a codificação *Cork*, chamada de `T1` no LaTeX. Essa codificação utiliza 8 bits, codificando assim 256 glifos. Foi projetada utilizando Metafont[1] e está disponível no formato *Adobe Type 1*[2] e também no formato *OpenType*[3].

A codificação de fonte pode ser escolhida utilizando-se o pacote `fontenc` (*font encoding*). Basta então adicioná-lo ao preâmbulo do documento. O exemplo na Listagem 3.10 ilustra a utilização da codificação Cork (T1).

Listagem 3.10: Especificando a codificação de fonte. Neste exemplo adotaremos a codificação Cork (T1).

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

3.5 Anatomia do documento

Um documento LaTeX é composto por diversas partes. Nesta seção definiremos e explicaremos cada parte do documento. A Listagem 3.11 apresenta um exemplo de um documento bem simples. A estrutura apresentada neste exemplo é o mínimo que todo documento deverá conter: definição da classe e corpo do documento.

Listagem 3.11: Documento bem simples.

[1] Metafont foi criada por Donald Knuth para ser uma linguagem de descrição para definir fontes raster.

[2] Adobe Type 1 é uma fonte PostScript desenvolvida pela Adobe.

[3] Formato de fontes escalonáveis desenvolvido pela Microsoft e Adobe

```
\documentclass{article}
\begin{document}
Teste!
\end{document}
```

A primeira linha define qual a classe de documento iremos utilizar, utilizando para tanto o comando `\documentclass`. Algumas classes usuais são: `article`, `report`, `book`, `letter`, `beamer`. A classe `article` utilizada no exemplo acima é designada para artigos em jornais científicos ou pequenos relatórios. Ao escolher a classe, estamos definindo qual é o arquivo de classe `.cls` que será utilizado no documento. Esse arquivo de classe define a estrutura básica e layout do documento. Você pode definir sua própria classe, mas este é um tópico mais avançado.

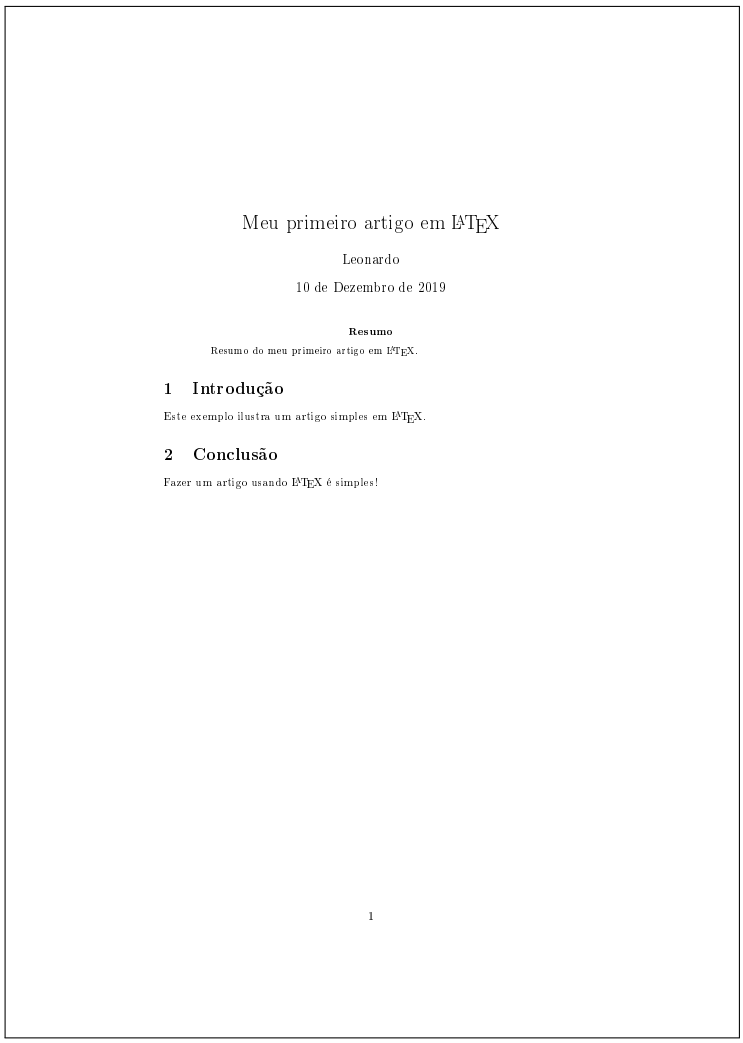
Após especificar a classe do nosso documento, definimos o corpo/conteúdo do documento dento da seção `document`. No exemplo da Listagem 3.11 adicionamos apenas um texto curto no documento: `Teste!`. Se você utilizar este código da Listagem 3.11 para compilar seu primeiro documento, verá que será produzida uma única página com o texto "Teste!" e o número da página na parte de baixo da folha.

Nosso primeiro exemplo foi muito simples e não continha muita coisa. Vamos agora passar para um exemplo mais ilustrativo, porém ainda simples. A Listagem 3.12 apresenta o código deste documento simples cujo resultado pode ser visualizado na Figura 3.5.

Listagem 3.12: Documento simples.

```
\documentclass[10pt,a4paper]{article}
\usepackage[utf8]{inputenc}
\usepackage[T1]{fontenc}
\usepackage[portuguese]{babel}
\author{Leonardo}
\title{Meu primeiro artigo em \LaTeX{}}
\begin{document}
\maketitle
\begin{abstract}
Resumo do meu primeiro artigo em \LaTeX{}.
\end{abstract}
\section{Introdução}
Este exemplo ilustra um artigo simples em \LaTeX{}.
```

```
14 \section{Conclusão}
15 Fazer um artigo usando \LaTeX{} é simples!
16 \end{document}
```



Meu primeiro artigo em LaTeX

Leonardo

10 de Dezembro de 2019

Resumo

Resumo do meu primeiro artigo em LaTeX.

1 Introdução

Este exemplo ilustra um artigo simples em LaTeX.

2 Conclusão

Fazer um artigo usando LaTeX é simples!

1

Figura 3.5: Exemplo de um documento simples.

Note na Listagem 3.12 que as estruturas apresentadas no primeiro exemplo também estão aqui presentes: definição da classe de documento e o corpo do documento. Na definição da classe ainda utilizamos dois argumen-

tos adicionais, especificando o tamanho da fonte `10pt` e o tamanho do papel `a4paper` (A4). Poderíamos ainda passar vários outros parâmetros, tais como `twocolumn` para produzir um texto com duas colunas, `twoside` para produzir um documento para ser impresso em frente-e-verso, `landscape` para gerar um documento no formato paisagem, `draft` para gerar um rascunho onde serão destacados os problemas de hifenização e justificação, `brazilian` poderá ser necessário em alguns casos para gerar os rótulos automáticos em português. Em seguida especificamos a codificação de entrada e codificação de fonte (como já explicamos nas seções 3.3 e 3.4, respectivamente). O pacote `babel` adiciona o suporte a várias línguas. No exemplo em questão selecionamos apenas o português, mas é possível selecionar outras línguas e mais de uma ao mesmo tempo, se o documento for multilíngue. Ainda no preambulo do documento, definimos o autor e o título, mas estas informações podem vir também dentro do corpo do documento, desde que venham antes do comando `\maketitle`.

Dentro do corpo do documento definimos o conteúdo que será exibido. Logo no início, chamamos o comando `\maketitle` que é responsável por imprimir o documento o título, autor e data. Note que a data não foi definida no preambulo do documento, desta forma será utilizada a dada de compilação do mesmo. Em seguida definimos a seção de resumo (*abstract*), onde iremos digitar o texto correspondente. O texto do nosso artigo inicia-se em seguida, sendo que estamos adotado a divisão em seções. Para definir cada uma das seções utilizamos o comando `\section` e passamos o nome da mesma.

Esta é uma estrutura bem simples de um documento, mas que já contempla muito do que será utilizado rotineiramente nos documentos preparados com LaTeX.

3.6 Pacotes

Muitas vezes iremos utilizar pacotes do LaTeX para termos acesso a certas funcionalidades que não estão disponíveis no sistema básico. Existem pacotes para as mais variadas funcionalidades. Já vimos até aqui a utilização de alguns deles: `inputenc`, `fontenc` e `babel`. Ao longo do livro veremos muitos outros. Sempre que você precisar fazer algo que não está disponível no LaTeX básico, é bem provável que alguém já implementou esta funcionalidade na forma de um pacote. Os pacotes ficam disponíveis no site do CTAN (`https://www.ctan.org/`). A instalação padrão do LaTeX já vem com diversos pacotes, o que é necessário para suprir a necessidade de grande parte do usuários. Caso você se depare com algum pacote indisponível em sua

instalação, proceda com a instalação do pacote necessário. Usualmente os pacotes do LaTeX vêem agrupados. Por exemplo, o pacote `texlive-science` no Ubuntu/Debian vem com 190 pacotes do CTAN agrupados. Mas se necessário você poderá baixar apenas o pacote desejado no site do CTAN e disponibilizar o arquivo juntamente com o seu documento.

3.7 Divisão do documento

Documentos costumam fazer a divisão dos tópicos em seções ou capítulos como uma forma de manter o conteúdo organizado. A organização do documento é muito importante para quem o redige e para quem o lê. As divisões também podem ser utilizadas para referenciar partes do texto, como veremos ainda adiante. Outra utilidade é a criação de sumários, que podem ser gerados automaticamente. Tópico que também veremos adiante.

A divisão do documento exemplo, na Listagem 3.12, foi feita em seções. Esta é a divisão usual em um artigo. Poderíamos ainda criar subseções dentro das seções e subsubseções dentro das subseções. Para tanto bastaria utilizar os comandos equivalentes.

Listagem 3.13: Usando seções para organizar o documento.

```
\section{Primeira seção}
\subsection{Sub-seção da primeira seção}
\section{Segunda seção}
\subsection{Sub-seção da segunda seção}
\subsubsection{Subsubseção}
```

No exemplo da Listagem 3.12, podemos verificar que as seções criadas já possuem numeração automática. Caso você queira suprimir a numeração, deverá fazer como ilustrado na Listagem 3.14.

Listagem 3.14: Criando uma seção sem numeração.

```
\section*{Seção sem numeração}
```

O LaTeX oferece 7 níveis de seccionamento do texto, nem todas serão utilizadas por todos documentos e raramente precisaremos utilizar vários níveis ao mesmo tempo. Como referimos acima, usualmente um artigo é dividido em seções e subseções. Em alguns raros casos utiliza-se subseções de subseções. Já um livro ou relatório costuma se dividido em capítulos, podendo ser

subdivididos em seções, subseções, etc. Os sete níveis mencionados são os seguintes expressos na Tabela 3.2. Os níveis `part` e `chapter` estão disponíveis apenas nas classes `report` e `book` .

Tabela 3.2: 7 níveis de organização do documento.

-1	\part{nome da parte}
0	\chapter{nome do capítulo}
1	\section{nome da seção}
2	\subsection{nome da subseção}
3	\subsubsection{nome da subsubseção}
4	\paragraph{nome do parágrafo}
5	\subparagraph{nome do subparágrafo}

3.8 Ambientes

Ambientes são definidos com `begin` e `end`, assim como foi feito para definir o corpo do documento: `\begin{document} ... \end{document}`. Existem vários outros ambientes padrões do LaTeX e muitos outros definidos nos diversos pacotes que pode ser utilizados. Os ambientes usualmente definem partes do documento que terão uma composição e exibição diferente do texto puro. Por exemplo, usamos ambientes para definir equações, tabelas, listas, citações, alinhamentos diferentes, etc. Ao longo do livro veremos vários exemplos e suas utilizações.

3.9 Tipografia das seções

Vimos até aqui o básico da estrutura de uma texto elaborado com LaTeX. Estabelecer esta estrutura é importante para a organização do texto. O LaTeX também cuida de fazer a numeração e gerar as referências para que as seções do nosso texto possam ser referenciadas de forma simples dentro dele, sem precisar nos preocupar com a numeração e sumário, uma vez que o LaTeX será capaz de gerar tudo de forma automática. Mas o objetivo de utilizar o LaTeX não é apenas organização e facilidades, queremos também gerar um documento que seja esteticamente agradável. Esta foi uma das principais preocupações de Knuth ao criar o TeX. O sistema TeX é capaz de controlar automaticamente o layout do nosso texto. Ao definir capítulos e seções, ele já o faz utilizando fonte de tamanho diferente, peso distinto e ainda controla os espaçamentos antes e depois. Tudo isso de forma automática, de forma

a criar um documento elegante e esteticamente bonito. O TeX controla automaticamente o espaçamento entre as letras e entre palavras, para evitar hifenizações e linhas com espaços grandes entre palavras formando buracos no meio do texto. Usando o LaTeX fica simples evitar seções com título perdido no final de uma página, ou sentenças órfãs do início de outra página. Caso o layout gerado de forma automática não lhe agrade em algum aspecto, você ainda pode ter controle sobre cada um dos parâmetros para ajustar e chegar no resultado desejado. É claro que estes são tópicos mais avançados e nem tudo será coberto neste livro.

3.10 Um exemplo de documento mais completo

Nesta seção apresentamos um exemplo de documento um pouco mais completo e utilizando alguns dos recursos disponíveis no LaTeX. O documento inicia com a definição da classe (no exemplo utilizamos a classe `book`), tamanho da fonte (`11`) e o tipo de papel (`A4`). Todas estas definições são passadas através do comando `\documentclass`. Depois são incluídos diversos pacotes para adicionar novas funcionalidades ao documento utilizando o comando `usepackage`. Os primeiros pacotes já foram abordados neste livro. Em seguida são inseridos pacotes matemáticos da distribuição AMS-LaTeX. Estes pacotes são utilizados para definir estruturas matemáticas e símbolos. Em textos que envolvam equações, teoremas e estruturas matemáticas será necessário utilizar estes pacotes. Inserimos também o pacote `graphicx` para inserirmos figuras e o pacote `lipsum` é utilizado no exemplo para gerar texto para preenchimento de documento.

Após isto define-se o título e iniciamos o ambiente documento (`begin`). Lembre-se de fechar o ambiente do documento ao final do texto (`end`).

O comando `\maketitle` é utilizado para apresentar o título no início do documento. Após isto é criado o sumário (`tableofcontents`) e a lista de figuras (`listoffigures`). Poderíamos ter criado também a lista de tabelas (`listoftables`).

Neste modelo `book` todos os capítulos foram numerados automaticamente de acordo com a ordem em que aparecem definidos (`chapter`). Poderíamos também utilizar capítulos não numerados (`chapter*`), seções (`section`) e subseções (`subsection`).

Depois definimos o estilo da bibliografia (`bibliographystyle`) e inserimos o ambiente de bibliografia em seguida (`thebibliography`). Outra forma mais prática costuma ser utilizar um arquivo `.bib` contendo a bibligrafia e carregá-lo no arquivo `.tex`. Veremos mais sobre isso no **??**. E por fim finalizamos o

documento (end).

Listagem 3.15: Exemplo de um documento simples em LaTeX.

```
\documentclass[11pt,a4paper,openany]{book}
% tipo de codificação
\usepackage[utf8]{inputenc}
% define idioma
\usepackage[portuguese]{babel}
\usepackage[T1]{fontenc}
% pacotes matemáticos
\usepackage{amsmath}
\usepackage{amsfonts}
\usepackage{amssymb}
% pacote gráfico
\usepackage{graphicx}
% gerar texto de preenchimento
\usepackage{lipsum}
\author{Alessandro Vivas \and Leonardo Araujo}
\title{Exemplo Simples}
\begin{document}
% cria o título
\maketitle
% cria o sumário
\tableofcontents
% lista de figuras
\listoffigures
\chapter{Introdução}
\lipsum[3-4]
\begin{figure}[t]
\centering
\includegraphics[width=4cm]{drawing.pdf}
\caption{Figura X.}\label{fig-drawing}
\end{figure}
\lipsum[5-6]
\chapter{Conclusões}
\lipsum[1-2]
\bibliographystyle{plain}
\begin{thebibliography}{9}
\bibitem{latexcompanion}
```

```latex
Donald E. Knuth.
\textit{The \TeX\ Book}.
Addison-Wesley, Massachusetts, 1984.
\end{thebibliography}
\end{document}
```

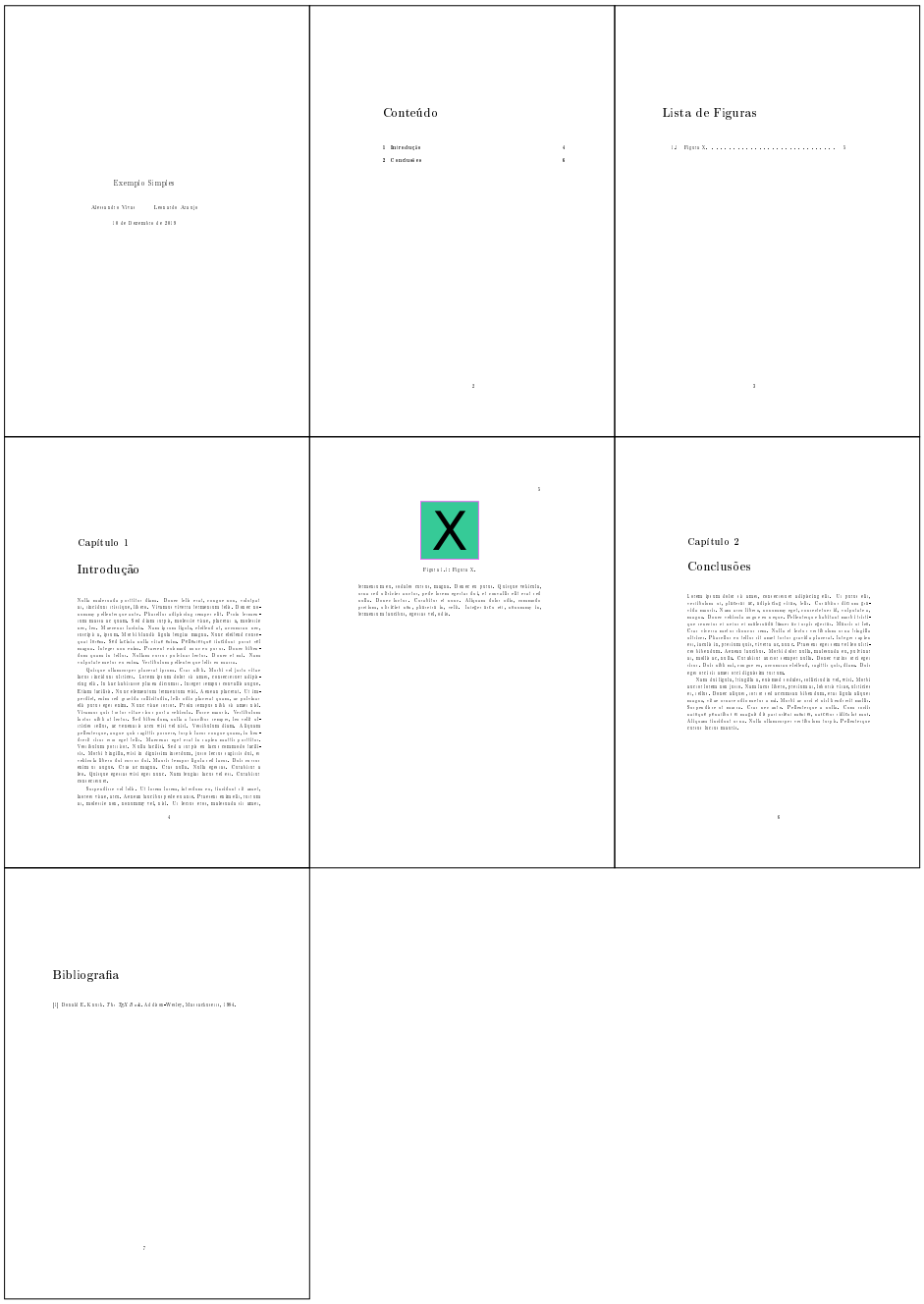


Figura 3.6: Exemplo do documento.

Capítulo 4

Texto

Sumário

4.1 Elemento básico de um documento 52
4.2 Formatação do texto 52
4.3 Espaçamento 53
4.4 Quebra de linha e página 56
4.5 Hifenização e agrupamentos 58
4.6 Alinhamento do texto 59
4.7 Citações e notas de rodapé 60
4.8 Acentos 61
4.9 Organização 62
4.10 Numeração 63

4.1 Elemento básico de um documento

O elemento básico de qualquer documento é texto. Já abordamos vários aspectos no capítulo inicial, como por exemplo: sintaxe básica do LaTeX, espaços, parágrafos e caracteres especiais na Seção 3.1; codificação para diferentes línguas na Seção 3.3; codificação de caracteres usada para escrever o arquivo `.tex` na Seção 3.3.1; e codificação de fonte do documento que será gerado pelo LaTeX na Seção 3.4. Mas ainda restam alguns aspectos e elementos de texto que serão tratados neste capítulo.

4.2 Formatação do texto

Vimos já que é possível escrever um texto com diferente tipo de ênfases como negrito, mas existem várias formas. Para ilustrá-las vamos apresentar na Listagem 4.1 alguns exemplos e os resultados de utilização.

Listagem 4.1: Alguns tipos de ênfase de fonte disponíveis.

```
% ênfase
\emph{texto teste 123} \\
% inclinada
\textsl{texto teste 123} \\
% itálico
\textit{texto teste 123} \\
% negrito
\textbf{texto teste 123} \\
% fonte romana
\textrm{texto teste 123} \\
% sem serifa
\textsf{texto teste 123} \\
% monoespaçada
\texttt{texto teste 123} \\
% maiúsculas menores
\textsc{texto teste 123} \\
% maiúsculas
\uppercase{texto teste 123}
```

texto teste 123
texto teste 123
texto teste 123
texto teste 123
texto teste 123
texto teste 123
texto teste 123
TEXTO TESTE 123
TEXTO TESTE 123

Como vimos na definição do documento na Seção 3.5, definimos o tamanho do texto normal na chamada ao comando `\documentclass`. Podemos definir tamanhos de texto relativo utilizando comandos que especificam o tamanho do texto. Veja os exemplos na Listagem 4.2.

Listagem 4.2: Variando o tamanho do texto.

texto texto texto texto texto texto

texto texto texto texto

```
\tiny{texto}
\scriptsize{texto}
\footnotesize{texto}
\small{texto}
\normalsize{texto}
\large{texto}
\Large{texto}
\LARGE{texto}
\huge{texto}
\Huge{texto}
```

Para a maioria dos casos utilizar os comandos da Listagem 4.2 será suficientes, mas, se desejar selecionar um tamanho de fonte arbitrário, isto poderá ser realizando conforme o exemplo da Listagem 4.3. Nesse exemplo especificamos uma fonte com tamanho de 3cm e uma espaçamento de 3.1cm.

Listagem 4.3: Fonte com tamanho arbitrário.

Teste

```
{
\fontsize{3cm}{3.1cm}
\selectfont
T\normalsize este
}
```

4.3 Espaçamento

Alguns textos utilizam padrão com espaçamentos diferentes. Usuamente utilizamos espaçamento simples, mas alguns padrões requerem espaçamento duplo ou espaçamento de um e meio. É possível alterar o espaçamento de um documeto utilizando o seguinte comando no preâmbulo:

Listagem 4.1: Comando para alterar o espaçamento do texto.

```
\linespread{fator}
```

Para ter um melhor controle, podemos utilizar também o pacote `setspace`, conforme o exemplo:

Listagem 4.2: Comando para alterar o espaçamento do texto.

```latex
\usepackage{setspace}
\singlespacing
%\onehalfspacing
%\doublespacing
```

Eventualmente pode ser que você precise adicional algum espaço na vertical ou na horizontal entre textos. Para este fim existem os comandos `\vspace` e `\hspace`, respectivamente. O exemplo da Listagem 4.4 apresenta a utilização desses dois comandos.

Listagem 4.4: Adicionando espaços na horizontal e vertical.

Teste Teste
123 456

789

```latex
Teste\hspace{5em}Teste\\
\vspace{3ex}123 456\\
\hspace{10em}789
```

Observação que pode ser útil: ambos comandos, `\hspace` e `\vspace`, também aceitam um espaçamento negativo.

No exemplo da Listagem 4.4 você deve ter notado que utilizamos unidades que talvez sejam desconhecidas para você: `em` e `ex`. Existem outras mais intuitivas como `mm` (milímetros), `cm` (centímetros), `in` (polegadas) e `pt` (pontos). Existem ainda outras, mas as citadas anteriormente são as mais convencionais. As duas unidades utilizada estão atreladas aos tamanhos dos tipos. A unidade `em` é a largura de um `M` maiúsculo, na fonte corrente e a unidade `ex` é a altura de um `x` minúsculo, na fonte corrente. A Tabela 4.1 apresenta o fator de conversão entre as unidades mencionadas. É importante ressaltar que em muitos casos é conveniente definir os espaços, tamanhos e distância de forma relativa a outros elementos no texto. Por exemplo, uma linha horizontal com 5cm de comprimendo pode ser adequada para um documento com uma coluna, mas passar a ser muito grande se o documento tiver duas colunas. Podemos então definir os valores relativos a outras referências, conforme veremos abaixo.

Uma forma util de utilizar o espaçamento sem fixar o valor é usar como comando `\fill`. Desta forma o espaço será feito de forma a preencher completamente o que falta na linha ou página (dependendo se estiver utilizando o `\hspace` ou `\vspace`, respectivamente). Como forma de simplificar, temos os atalhos: `\hfill` para designar `\hspace{\fill}` e `\vfill` para designar `\vspace{\fill}`.

Tabela 4.1: Fatos de conversão entre as unidades (tabela adaptada do stackexchange).

1pt	**1mm**	**1cm**	**1ex**	**1em**	**1in**	
1	2.84526	28.45274	4.71341	10.88788	72.26999	**pt**
0.35146	1	10.00005	1.65659	3.82668	25.40013	**mm**
0.03514	0.1	1	0.16566	0.38266	2.54	**cm**
0.21216	0.60365	6.03656	1	2.30998	15.33286	**ex**
0.09184	0.26132	2.61325	0.4329	1	6.63765	**em**
0.01384	0.03937	0.3937	0.06522	0.15065	1	**in**

Além de poder definir os espaçamentos através das unidades padrão aceitas pelo LaTeX (cm, in, pt), existem alguns espaçamentos pré-definidos que costumam ser úteis. Alguns deles são apresentados na lista abaixo:

`\baselineskip` é o espaçamento entre duas linhas dentro de um parágrafo.

`\columnsep` é a distância entre duas colunas.

`\columnwidth` é a largura de uma coluna.

`\linewidth` é a largura de uma linha de texto no ambiente local.

`\paperwidth` é a largura do papel.

`\paperheight` é a altura do papel.

`\textheight` é a altura do texto.

`\textwidth` é a largura do texto.

Qualquer um desses também aceitam múltiplos ou frações, basta para tanto anteceder o espaçamento desejado pelo fator multiplicativo. Por exemplo, veja a Listagem 4.3.

Listagem 4.3: Utilizando as unidades pré-definidas no LaTeX.

```
1 \includegraphics[width=0.5\textwidth]{mygraphic}
```

4.4 Quebra de linha e página

O LaTeX costuma fazer um bom trabalho fazendo as quebras de linhas e página em locais apropriados, deixando o documento bem estruturado e bonito. Entretanto, podem ocorrer situações em que o LaTeX não fará a melhor escolha, ou você pode preferir alguma quebra diferente daquela gerada pelo LaTeX. Como o texto está em constante mudança, o que acarretará na mudança da escolha dos pontos de quebra linha ou página, é melhor fazer os ajustes necessários apenas no final, quando o documento já estiver pronto e não sofrerá mais edições.

Podem ocorrer situações em que o LaTeX não encontra um local apropriado para quebrar uma linha, deixando espaços grandes ou, o que pode ser ainda pior, algum texto avançando além da margem. Também podemos nos deparar com casos em que o LaTeX realiza uma quebra de página deixando órfãos, o que muitas vezes não nos agrada muito. Todos os problemas podem ser solucionados ou contornados.

O problema em quebra de linha pode ser pelo fato do LaTeX não conseguir hifenizar uma palavra. Veja na Seção 4.5 como definimos a hifenização de palavras, o que pode ser útil para lavras que ocorram frequentemente e que o LaTeX tenha dificuldades em hifenizar.

Se um determinado parágrafo não resultou em boas quebras de linhas, uma possibilidade é tentar reescrevê-lo. Às vezes, pequenas mudanças podem fazer com que o LaTeX consiga gerar boas quebras de linhas. Caso as palavras e ordem escolhidas tenham sido minuciosamente escolhidas, não sendo adequado qualquer alteração, podemos pedir ao LaTeX para ser um pouco mais desleixado neste parágrafo. Isto pode ser feito colocando o parágrafo em questão dentro do ambiente `sloppypar`, conforme ilustrado na Listagem 4.4.

Listagem 4.4: Fazendo o LaTeX ser menos meticuloso.

```
\begin{sloppypar}
Neste parágrafo o \LaTeX{} será menos meticuloso
ao gerar os espaços entre letras e palavras, possibilitando
maior maleabilidade com o fim de conseguir melhorar a alocação
de quebras de linhas.
\end{sloppypar}
```

Alternativamente podemos utiliza o comando `\sloppy`, o que deixará o LaTeX menos meticuloso a partir deste ponto. Caso opte por esta opção, é interessante restringir o desleixo apenas a uma parcela do documento man-

tida entre chaves ou restaurando a configuração compulsiva padrão com o comando `\fussy`, conforme ilustrado na Listagem 4.5.

Listagem 4.5: Outra forma de fazer o LaTeX ser menos meticuloso.

```
{
  \sloppy
  Dentro das chaves o \LaTeX{} será menos meticuloso.
}

(...)

\sloppy
Neste trecho o \LaTeX{} será menos meticuloso.
\fussy
Agora volta ao normal.
```

Em último caso, você ainda poderá forçar uma quebra de linha no local desejado utilizando o comando `\linebreak`. Este comando aceita como parâmetro um número de 0 a 4, devendo ser passado entre colchetes `\linebreak[n]`. O parâmentro 0 apenas sugere ao LaTeX que aquele local seria um bom local para fazer uma quebra de linha e 4 força a quebra de linha no local. O comportamento padrão, quando não passamos um número como parâmetro, é equivalente a passar o número 4 como parâmetro.

O comando `\linebreak` faz com que o TeX justifique a linha, enquanto o comando `\newline` (equivalente ao `\\`) termina a linha sem promover a justificação.

Os comandos do LaTeX para quebra de página são análogos aos comandos para quebra de linha. Podemos utilizar os comandos `\pagebreak` e `\nopagebreak`, que da mesma forma aceitam como parâmetro um número de 0 a 4 para fazer uma gradação entre sugerir e obrigar a quebra ou não quebra de página em um determinado ponto do documento.

De forma análoga temos também o comando `\newpage` para terminar uma página prematuramente e começar uma nova. O comando `\clearpage` é similar, deixando qualquer figura ou tabela remanescente em uma página vazia sem texto.

4.5 Hifenização e agrupamentos

O LaTeX faz hifenização automática, quando necessário. Como hifenizar palavras prejudica a leitura, hifenização será evitada, a menos que seja a única solução sem comprometer a estética do texto. Para a hifenização funcionar adequadamente, lembre-se de selecionar a língua correta do documento. Existe um algoritmo de hifenização que varia de acordo com a língua selecionada. Esse algoritmo pode errar, então, caso isso ocorra, você deverá especificar explicitamente como a palavra que causou erro deve ser hifenizada. Para tanto, no preâmbulo do documento utilize o comando `hyphenation` especificando explicitamente a hifenização das palavras, conforme o exemplo na Listagem 4.6.

Listagem 4.6: Comando para alterar o espaçamento do texto.

```
1 \hyphenation{pa-la-vra,e-xem-plo}
```

Em algumas situações temos textos que, mesmo separados por espaços em branco, desejamos que sejam mantidos juntos, evitando que sejam separados por uma quebra de linha quando aparecerem ao final da linha. Uma possibilidade é especificar quais são os espaços onde não queremos permitir uma quebra de linha. Por exemplo, no nome *Leonardo C. Araújo*, não queremos que o *C.* apareça no início da linha, para tanto ao invés de usar espaço utilizaremos o caractere `~`. Ou ainda, podemos especificar que uma determinada sequência de palavras deva permanecer coesa sem quebra de linha, sequer hifenização. Para tanto usamos o comando `\mbox`. Veja o exemplo da Listagem 4.5.

Listagem 4.5: Mantendo um texto coeso.

Um dos autores do livro: Leonardo C. Araújo
Um dos autores do livro: Leonardo C. Araújo
Um dos autores do livro: Leonardo C. Araújo

```
1 Um dos autores do livro:
    Leonardo C. Araújo
2
3 Um dos autores do livro:
    Leonardo~C. Araújo
4
5 Um dos autores do livro: \
    mbox{Leonardo C. Araújo}
```

O comando `\mbox` é uma abreviação para `\makebox` que, como o nome sugere, cria uma caixa com o conteúdo que foi passado como parâmetro, de

forma semelhante às caixas utilizadas para agrupar tipos móveis das prensas mecânicas. As caixas criadas pelo TeX possuem pontos de referência na lateral esquerda. Para criar linhas o TeX coloca caixas lado a lado alinhando seus pontos de referência. A Figura 4.1 mostra o diagrama de duas caixas, evidenciando o ponto de alinhamento. O TeX cria as caixas para cada letra, como se fossem um tipo gráfico virtual e adiciona espaço variável entre elas (*glue*). O exemplo da Figura 4.1 ilustra o caso em que se cria uma caixa como resultado do comando `\mbox{pnat}`.

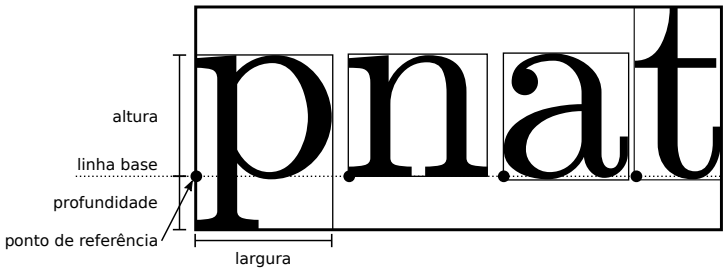


Figura 4.1: Como o TeX faz a organização do texto utilizando caixas.

4.6 Alinhamento do texto

O LaTeX possui ambientes para alinhamento do texto centralizado, à direita e à esquerda. São eles, respectivamente: `center`, `flushright` e `flushleft`. A Listagem 4.6 apresenta exemplo de utilização desses ambientes.

Listagem 4.6: Utilização dos alinhamentos de texto.

```
\begin{center}
Este texto ficará centralizado
    na página.
\end{center}

\begin{flushright}
Este à direita.
\end{flushright}

\begin{flushleft}
Este à esquerda.
\end{flushleft}
```

Este texto ficará centralizado na página.

Este à direita.

Este à esquerda.

Também podemos utilizar os respectivos comandos (`\centering`, `\raggedright`, `\raggedleft`) para promover o alinhamento em uma região delimiatada por chaves ou dentro de um ambiente.

4.7 Citações e notas de rodapé

Quando utilizamos em um documento um texto que não é de nossa autoria, devemos marcar explicitamente isso. Para tanto existem diferentes formas, que variam com a língua adotada. A Listagem 4.7 apresenta algumas formas possíveis de marcar usando símbolos.

Listagem 4.7: Citações em um texto.

```
Descartes disse `Cogito, ergo sum.'

Descartes disse ``Cogito, ergo sum.''

Descartes disse ,,Cogito, ergo sum.''

Descartes disse <<Cogito, ergo sum.>>
```

Descartes disse ‘Cogito, ergo sum.’
Descartes disse “Cogito, ergo sum.”
Descartes disse „Cogito, ergo sum.”
Descartes disse «Cogito, ergo sum.»

Quando queremos citar um trecho mais longo, usualmente utilizamos o ambiente `quote` para exibir o texto com espaçamento da margem diferenciado. Na Listagem 4.7 apresentamos um exemplo de uma citação de Machado de Assim, onde o seu resultado pode ser visto a seguir.

Listagem 4.7: Exemplo de uma citação de um trecho maior.

```
\begin{quote}
Palavra puxa palavra, uma idéia traz outra, e assim se faz um
livro, um governo, ou uma revolução. (Machado de Assis\footnote{
Machado de Assis é considerado um dos maiores nomes da
Literatura Brasileira.})
\end{quote}
```

> Palavra puxa palavra, uma idéia traz outra, e assim se faz um livro, um governo, ou uma revolução. (Machado de Assis[1])

Outro recurso que é recorrente em textos é a utilização de notas de rodapé que são feitas através do comando `\footnote`. O padrão ABNT adota um padrão singular onde as nota de rodapé também são utilizadas para referências bibliográficas. Não se preocupe com isso agora, você não irá utilizar o comando `\footnote` para fazer as notas de rodapé diretamente. Existe um pacote para LaTeX que converte as referências automaticamente para o padrão ABNT, o que não é assunto para esta seção. Nos preocuparemos aqui apenas com as notas de rodapé para fornecer informações adicionais sobre algo do nosso texto. No exemplo na Listagem 4.7 apresentamos um exemplo de utilização do comando `\footnote`, responsável por gerar a nota de rodapé desta página.

4.8 Acentos

O LaTeX é capaz de produzir textos utilizando as mais variadas línguas e seus distintos alfabetos, entretanto nosso foco aqui é produzir textos em português e inglês. Desta forma iremos nos restringir ao que será necessário: acentos.

Se estiver escrevendo o texto `.tex` utilizando uma codificação que suporte acentos e colocar os parâmetros corretos no cabeçalho, como vimos na Seção 3.3, os acentos digitados no arquivo `.tex` apareceram no texto final gerado pelo LaTeX sem mistério algum. Recomendamos a utilização do UTF-8, que tornou-se a língua franca das codificações com a ubiquação da internet. Caso esteja utilizando uma codificação que não suporte acentos, a codificação ASCII por exemplo, ou caso deseje que seu `.tex` seja compatível com máquinas antigas sem suporte a codificações que aceitem acentuação, podemos definir os acentos apenas utilizando os caracteres ASCII. Devemos utilizar o acento desejado precedido por contrabarra e, em seguida, entre chaves o caractere que receberá o acento. Veja alguns exemplos na Listagem 4.8.

Listagem 4.8: Como utilizar acentos em sistemas que não suportam acentos.

```
1 Amanh\~{a} o gar\c{c}on
2 ir\'{a} servir
3 ling\"{u}i\c{c}a.
```

Amanhã o garçon irá servir lingüiça.

[1] Machado de Assis é considerado um dos maiores nomes da Literatura Brasileira.

4.9 Organização

Na elaboração de um documento, sobretudo de um documento grande, é muito importante organizar bem as informações (textos, tabelas, figuras, equações, etc.) que farão parte deste documento. A organização é importante para o leitor, mas crucial para o escritor. Usando o LaTeX podemos dividir e organizar essas informações de forma a facilitar a elaboração, edição e revisão do documento.

Podemos utilizar como estratégia a separação do documento em diversas partes, sendo que estas serão juntadas utilizando o comando `\input`. Podemos, por exemplo, salvar cada capítulo ou seção de um livro ou documento em arquivos `.tex` separados. Teremos então um arquivo `.tex` principal no qual faremos a inclusão dos demais utilizando o comando `\input`. Veja o exemplo da Listagem 4.8 onde apresentados o código do arquivo principal `livro.tex`.

Listagem 4.8: Exemplo de utilização do comando `input`. Arquivo `livro.tex`.

```
1 \documentclass[a4paper,11pt]{book}
2 \begin{document}
3 \tableofcontents
4 \input{capitulo1.tex}
5 \input{capitulo2.tex}
6 \input{capitulo3.tex}
7 \input{capitulo4.tex}
8 \end{document}
```

O comando `\input` simplesmente insere o conteúdo do arquivo passado como parâmetro no ponto em que o comando foi chamado. Seria como se, no ponto onde é feito o `\input`, fizéssemos a colagem do conteúdo do arquivo referenciado como parâmetro do `\input`. Desta forma podemos separar conteúdos em arquivo específicos, como no exemplo da Listagem 4.8, onde cada capítulo está em um arquivo separado. Caso tenhamos uma tabela grande, podemos deixá-la em um arquivo `.tex` separado. Isto torna prático gerar o documento final com algumas partes ou alterar partes específicas. Suponha que você faça análise de dados no `R`, poderá utilizar o pacote do `R` `stargazer` paga gerar as tabelas. Salvando-as em um arquivo independente fica fácil substituí-las caso encontre algum erro nas análises e precise gerar as tabelas novamente.

Utilizando `\input` fica simples gerar uma versão do livro em que terá

apenas um ou dois capítulos, ou uma versão onde iremos suprimir determinada tabela que veio a ocasionar algum problema. Basta comentar a linha com `\input` desejada, resolver o que for necessário e depois reinserir o que foi retirado apenas removendo o comentário.

Note que o `\input` irá apenas inserir o conteúdo de um arquivo no local onde foi chamado o comando. Desta forma, se o arquivo inserido possuir uma imagem, por exemplo, o caminho desta imagem poderá gerar algum problema se o primeiro arquivo onde foi feito o `\input` e o arquivo de imagem estiverem em locais diferentes. Utilizar sempre o caminho completo pode ser uma forma de evitar estes problemas, ou então utilizar o pacote `import` que fará as adequações de caminho dos arquivos para não ocorrer erro em localizar os arquivos inseridos.

4.10 Numeração

O LaTeX faz numeração automática dos elementos que constituem o texto: capítulos, seções, páginas, equações, figuras, tabelas, etc. Para tanto ele utiliza-se de contadores com nome, em geral, igual ao nome do ambiente ou comando que produz o elemento que será numerado. Os valores dos contadores são inteiros não negativos, sendo automaticamente incrementados todas vez que uma nova instância de um ambiente é criado no documento ou toda vez que o comando respectivo é utilizado.

Usualmente não é necessário trabalhar diretamente com os valores dos contadores, uma vez que o LaTeX faz tudo de forma automática. Se for necessários podemos acessar e alterar os valores dos contadores usando os comandos `\value`, `\setcounter` e `\addtocounter`, para ver o valor, alterar o valor ou adicionar ao valor do contador.

A Listagem 4.9 ilustra o caso de alteração do valor do contador de notas de rodapés.

Listagem 4.9: Exemplo de manipulação direta dos contadores.

```
\setcounter{footnote}{11}
texto texto\footnote{ ... }
\addtocounter{footnote}{-1}
texto texto\footnote{ ... }
```

O LaTeX permite imprimirmos contadores em diferentes formatos: algarismos arábicos (`\arabic`), algarismos romanos (`\roman` ou `\Roman`) ou

alfabético (`\alph` ou `\Alph`). Veja na Listagem 4.9 o exemplo de utilização para imprimir o número da página corrente nas diversas formas.

Listagem 4.9: Exemplos de formatos de impressão dos contadores.

```
page \arabic{page}, footnote \
    roman{footnote}, table \
    Roman{table},
chapter \alph{chapter}, section
     \Alph{section}
```

page 64, footnote i, table I, chapter d, section J

Capítulo 5

Trabalhando com Figuras

Sumário

5.1 Ambiente `figure` 66
5.2 Opções de posicionamento 66
5.3 Rotacionando figuras 69
5.4 Alterando o altura, largura e escala 71
5.4.1 Alterando a largura 71
5.4.2 Alterando a altura 72
5.4.3 Alterando a escala 73
5.5 Cortando a figura 74
5.6 Opções conjuntas 75
5.7 Inserindo várias figuras 75

5.1 Ambiente `figure`

Para incluir imagens em documentos podemos usar alguns pacotes, mas o mais usual é o `graphicx`. Incluímos então no preâmbulo a linha (antes do comando `\begin{document}`) `\usepackage{graphicx}`.

O processo de inserção de figuras inicia com a definição do ambiente `figure` da seguinte maneira: `\begin{figure}[posicionamento]...\end{figure}`.

As figuras podem ficar em qualquer pasta do disco rígido, porém, ao serem inseridas, devemos especificar o caminho completo ou relativo de onde elas se encontram. No exemplo a seguir as figuras estão armazenadas no diretório imagens. Caso a figura esteja no mesmo diretório do arquivo basta colocar o nome da figura, mas se estiver em outro local terá que especificar o caminho.

Você poderá definir várias maneiras de posicionamento, no parâmetro entre os colchetes (`[ ]`) do ambiente `figure`. Também poderá definir tamanho, escala e rotação nos parâmetros do comando `\includegraphics`.

A Listagem 5.1 apresenta o procedimento de inserir uma figura no LaTeX. Note que o ambiente da figura foi definido com o parâmetro `h` (*here*, veja Seção 5.2) e o comando `\includegraphics` utilizou o parâmetro `width=10cm`, definindo a largura da imagem.

Listagem 5.1: Inserir Figuras no LaTeX

```
\begin{figure}[h]
        \center
        \includegraphics[width=10cm]{./imagens/exemplo01.jpg}
        \caption{Legenda da Imagem}
        \label{label}
\end{figure}
```

A Figura 5.1 apresenta o resultado da Listagem 5.1.

5.2 Opções de posicionamento

Existem algumas maneiras de alterar o posicionamento de uma figura, dentre elas destacam-se:

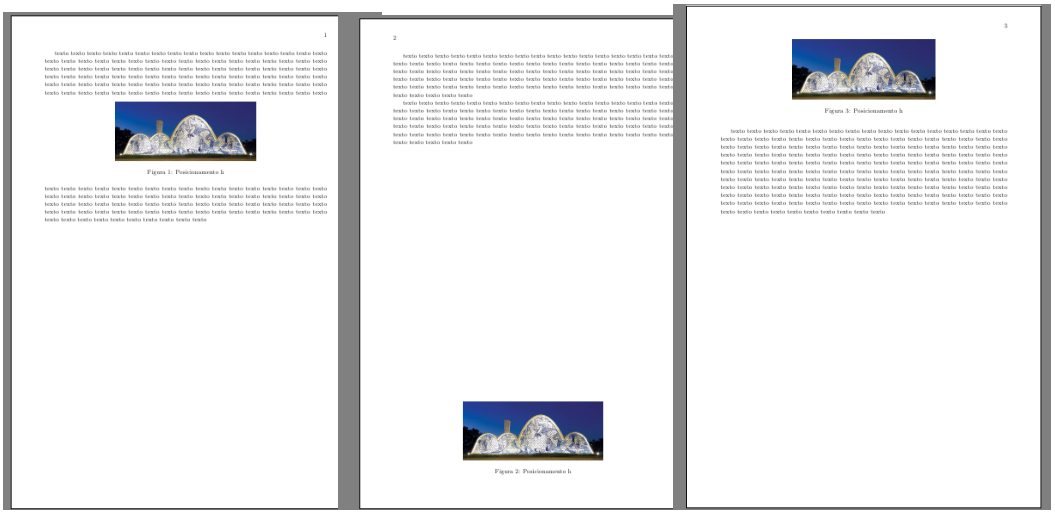
- h - para que a imagem fique exatamente na parte do texto em que é introduzida (here);

Figura 5.1: Legenda da Imagem

- t - para a imagem aparecer no topo da página (top);
- b - para a imagem aparecer no fundo da página (bottom);
- p - para a imagem aparecer numa página só com figuras ou tabelas.

Note que em todos os casos, a opção de posicionamento escolhida será apenas uma indicação do que desejamos que o LaTeX faça o posicionamento da figura. Entretanto, se esta escolha gerar linhas malformadas, grandes espaços vazios, linhas órfãs, dentre outros problemas, o LaTeX poderá trocar a escolha de posicionamento. O padrão do LaTeX é deixar que as figuras e tabelas flutuem no texto, de forma que ele possa buscar o melhor posicionamento delas. Isto significa que em um texto você deverá evitar fazer referências a elas na forma: `como podemos ver na figura abaixo`. Deverá buscar sempre referenciar figuras e tabelas pelos seus nomes, como no seguinte exemplo: `como podemos ver na Figura \ref{label-da-figura}`. Você poderá tentar dar ênfase da sua escolha de posicionamento da figura adicionado um exclamação, por exemplo, `\begin{figure}[h!]...\end{figure}`. Se isto não resolver e se você estiver realmente incomodado com a disposição das figuras e tabelas, poderá tentar utilizar o pacote `float`, utilizando então a opção `H` para ditar o posicionamento.

A Figura 5.2 mostra três maneiras de posicionar as figuras: a) utilizando `\begin{figure}[h]`, b) ilustra o posicionamento utilizando `\begin{figure}[t]`, e c) ilustra o posicionamento utilizando `\begin{figure}[b]`.



(a) Opção h (b) Opção t (c) Opção b

Figura 5.2: Opções de posicionamento de Figuras

A Listagem 5.2 apresenta o código que gera a Figura 5.2. Para este código funcionar é necessário incluir no início do documento o pacote `subfigure`.

Listagem 5.2: Posicionamento das Figuras

```
\begin{figure}[h]
\begin{subfigure}[b]{0.3\textwidth} \centering
\includegraphics[width=5cm]{Figuras/posicionamento01.png}
\caption{Opção h}\label{fig:posicionamento01}
\end{subfigure}
\hfill
\begin{subfigure}[b]{0.3\textwidth} \centering
\includegraphics[width=5cm]{Figuras/posicionamento02.png}
\caption{Opção t}\label{fig:posicionamento02}
\end{subfigure}
\hfill
\begin{subfigure}[b]{0.3\textwidth} \centering
\includegraphics[width=5cm]{Figuras/posicionamento03.png}
\caption{Opção b}\label{fig:posicionamento03}
\end{subfigure}
```

```
\caption{Opções de posicionamento de Figuras}
\label{posicionamentofiguras}
\end{figure}
```

5.3 Rotacionando figuras

Para girar uma figura em LaTeX não é necessário alterar a imagem, pois a rotação é feita ao se compilar o documento. Você necessita apenas inserir o comando de rotação:

Listagem 5.3: Rotacionando Figuras

```
\includegraphics[angle=Graus]{·}
```

A Figura 5.3 ilustra a rotação de figuras nos seguintes ângulos: 15, 30, 45 e 60 graus.

O código para rotacionar as figuras está na Listagem 5.4.

Listagem 5.4: Rotacionando Figuras

```
\begin{figure}[h]
\begin{subfigure}[b]{0.25\textwidth}
\centering
\includegraphics[scale=0.02,angle=15]{Figuras/flores02.jpg}
\caption{}  \label{fig-flor15}
\end{subfigure}
\begin{subfigure}[b]{0.25\textwidth}
\centering
\includegraphics[scale=0.02,angle=30]{Figuras/flores02.jpg}
\caption{}  \label{fig-flor30}
\end{subfigure}
\begin{subfigure}[b]{0.25\textwidth}
\centering
```

(a)

(b)

(c)

(d)

Figura 5.3: Opções Rotação

```
\includegraphics[scale=0.02,angle=45]{Figuras/flores02.jpg}
\caption{}  \label{fig-flor45}
\end{subfigure}
\begin{subfigure}[b]{0.25\textwidth}
\centering
\includegraphics[scale=0.02,angle=60]{Figuras/flores02.jpg}
\caption{}  \label{fig-flor60}
\end{subfigure}
\caption{Opções Rotação}
\label{rotacaodefiguras}
\end{figure}
```

5.4 Alterando o altura, largura e escala

Em LaTeX a figura pode ser armazenada em seu tamanho original e podemos controlar seu tamanho através das opções do ambiente `figure`. Existem três maneiras de definir o tamanho de uma figura:

- controlando a altura `\includegraphics[height=60mm]{figura01.png}`;
- controlando a largura `\includegraphics[width=60mm]{figura01.png}`;
- controlando a escala percentual `\includegraphics[scale=0.35]{figura01.png}`.

5.4.1 Alterando a largura

A Figura 5.4 ilustra a opção de alteração da largura da figura:

(a) 3 cm (b) 5 cm

Figura 5.4: Trabalhando com a Largura

O código para alterar a largura da figura é apresentando na Listagem 5.5.

Listagem 5.5: Alterando a Largura

```
\begin{figure}[h]
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[width=3cm]{Figuras/flores02.jpg}
\caption{3 cm} \label{flor3cm}
\end{subfigure}
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[width=5cm]{Figuras/flores02.jpg}
\caption{5 cm} \label{flor5cm}
\end{subfigure}
\caption{Trabalhando com a Largura}
\label{largurafigura}
\end{figure}
```

5.4.2 Alterando a altura

Nesta opção você define a altura da figura que está trabalhando. Vamos utilizar dois exemplos na mesma figura, no primeiro exemplo definimos uma altura de 1cm e no segundo exemplo definimos uma altura de 2cm. Para definir a altura utilizamos o parâmetro `height` seguido do valor desejado. A Figura 5.5 ilustra a opção de alteração da altura da figura:

(a) 1 cm (b) 2 cm

Figura 5.5: Trabalhando com a altura

A Listagem 5.6 apresenta o código para alterar a altura como na Figura 5.5.

Listagem 5.6: Alterando a altura

```
\begin{figure}[h]
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[height=1cm]{Figuras/flores02.jpg}
\caption{1 cm} \label{florh1cm}
\end{subfigure}
```

```
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[height=2cm]{Figuras/flores02.jpg}
\caption{2 cm} \label{florh2cm}
\end{subfigure}
\caption{Trabalhando com a Altura}
\label{alturafigura}
\end{figure}
```

5.4.3 Alterando a escala

Para alterar a escala da figura utilizamos o parâmetro `scale`. A Figura 5.6 ilustra a opção de alteração da escala da figura.

(a) 5% (b) 2%

Figura 5.6: Trabalhando com a escala

A Listagem 5.7 produz a Figura 5.6.

Listagem 5.7: Alterando a Escala

```
\begin{figure}[!h]
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[scale=0.05]{Figuras/flores02.jpg}
\caption{5\%} \label{figflor5pp}
\end{subfigure}
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[scale=0.02]{Figuras/flores02.jpg}
\caption{2\%} \label{figflor2pp}
```

```
\end{subfigure}
\caption{Trabalhando com a escala}
\label{escalafigura}
\end{figure}
```

5.5 Cortando a figura

Se precisar mostrar algum detalhe de uma figura não é necessário utilizar algum software gráfico e fazer cortes no mesmo. É possível utilizar opções para mostrar apenas uma parte da figura e para isto utilizamos a opção `trim`. Esta opção funciona definindo 4 parâmetros:

- left : distância em relação a esquerda da figura;
- botton: distância em relação a parte inferior da figura;
- right: distância em relação a parte direita da figura;
- top: distância em relação a parte superior da figura.

A Figura 5.7 apresenta o código para gerar a Figura 5.7.

Listagem 5.8: Cortando a Figura

```
\begin{figure}[h]
\center
\includegraphics[width=10cm,trim=10mm 30mm 10mm 30mm, clip]
    {./figuras/f02.jpg}
\caption{Cortando uma Figura}
\label{figuratrim}
\end{figure}
```

(a) Original

(b) Utilizando Trim

Figura 5.7: Cortando a Figura

5.6 Opções conjuntas

O LaTeX permite que você utilize diversas opções em conjunto facilitando o trabalho de criação do documento. Vamos supor agora que você quer que sua figura fique no topo da página, com uma escala de 25% e ângulo de 75 graus. A Listagem 5.9 apresenta o código para gerar a Figura 5.8.

Listagem 5.9: Opções Conjuntas

```
\begin{figure}[h]
\center
\includegraphics[scale=0.25,angle=75]{./figuras/f02.jpg}
\caption{Opções Compostas}
\label{opcoescompostas}
\end{figure}
```

5.7 Inserindo várias figuras

Para trabalhar com várias figuras simultâneas basta utilizar o pacote `subfigure`:

Listagem 5.10: Pacote subfigure.

```
\usepackage{subfigure}
```

Podemos utilizar este pacote para mostrar diversas figuras com legendas. A Listagem 5.11 apresenta o código para gerar a Figura 5.9. A Figura 5.9a apresenta a figura com escala de 10% e a Figura 5.9b apresenta com escala de 5%.

Figura 5.8: Opções Compostas

Listagem 5.11: Várias Figuras

```
\begin{figure}[h]
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[scale=0.10]{Figuras/flores02.jpg}
\caption{Exemplo 1}\label{l1a}
\end{subfigure}
\begin{subfigure}[b]{0.5\textwidth}
\centering
\includegraphics[scale=0.05]{Figuras/flores02.jpg}
\caption{Exemplo 2}\label{l2a}
\end{subfigure}
\caption{Várias Figuras}
\label{figurasubfigure}
\end{figure}
```

(a) Exemplo 1 (b) Exemplo 2

Figura 5.9: Várias Figuras

Capítulo 6

Tabelas

Sumário

6.1 Tabelas **80**
6.2 Elementos básicos **80**
6.3 Tabelas longas **84**

6.1 Tabelas

As tabelas são estruturas na forma de grade usadas para organizar informações, tornando-as visualmente acessível ao leitor. A maneira escolhida para organizar as informações em uma tabela podem explicitar certos dados e facilitar a leitura de um texto, tornando-se objetos muito importantes na organização de um texto. É indispensável que o autor primeiramente tenha em mente quais dados deseja exibir em uma tabela e como deseja organizar a informação. O LaTeX possui ferramentas especificas para construir tabelas de todos os tipos. Saber usá-las é indispensável para a produzir um bom texto.

Tabelas mais simples são facilmente implementadas em LaTeX. Já aquelas mais elaboradas e rebuscadas requerem um pouco mais de prática e pacotes adicionais. O LaTeX permite criar qualquer tipo de tabela e provavelmente já existe algum pacote para suprir alguma necessidade que você venha a ter ao criar tabelas. É possível criar tabelas de tamanhos diferentes; alterar os alinhamentos dos elementos; expandir células; colorir linhas, colunas e células; rotacionar; carregar dados de um arquivo externo (`.csv`, por exemplo); gerar tabelas através de códigos; e muito mais. Para cada funcionalidade você deverá buscar o pacote necessário para suprir suas necessidades. Todos pacotes possuem documentação no CTAN, basta consultá-la para aprender a utilizar os pacotes e ver exemplos de utilização. Um dos pacotes principais para criar tabelas mais elaboradas é o `tabu`. Outros pacotes disponíveis e comumente utilizados são o `longtable` e o `tabularx`. Vamos nos ater inicialmente aos elementos básicos na próxima secção.

6.2 Elementos básicos

O ambiente básico para composição de tabelas é o `tabular`. Assim como os demais, usa-se o `begin` e `end` para iniciar e terminar este ambiente. Ao criar este ambiente passamos um argumento como parâmetro especificando a definição das colunas, o alinhamento das colunas, dado por uma letra, e assim, implicitamente também definimos o número de colunas pelo número de letras no argumento. Vejamos um primeiro exemplo em que iremos organizar os números de 1 a 9 em uma tabela 3×3.

Listagem 6.1: Primeiro exemplo simples de tabela.

1	2	3
4	5	6
7	8	9

```
\begin{tabular}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{tabular}
```

Note que neste primeiro exemplo, ao definir o ambiente `tabular` passamos como parâmetro `ccc`, indicando que as três colunas terão alinhamento centralizado (`c`). As outras opções possíveis para o alinhamento são: à esquerda (`l`), à direita (`r`) e parágrafo (`p`, `m` ou `b`). [1] Observe no exemplo acima que os dados que irão popular as colunas são separados pelo caractere `&`, já as linhas em uma tabela são separadas pela quebra de linha `\\`. No parâmetro de definição das colunas, devemos informar também as divisões através de linhas verticais, se forem desejadas, utilizando para tanto o caractere `|` (barra vertical). Já as linhas horizontais são definidas com o comando `\hline`. Vamos refazer o exemplo anterior acrescentando as divisões por linhas verticais e horizontais.

Listagem 6.2: Exemplo simples de tabela com divisões.

1	2	3
4	5	6
7	8	9

```
\begin{tabular}{|c|c|c|}
\hline
1 & 2 & 3 \\ \hline
4 & 5 & 6 \\ \hline
7 & 8 & 9 \\ \hline
\end{tabular}
```

Muitos editores LaTeX possuem ferramentas para facilitar e automatizar o processo de criação de tabelas. Caso seu editor não ofereça, existem ferramentas disponíveis na internet que também iráo auxiliar.

Vamos agora fazer um exemplo em que uma das colunas conterá um texto longo. Para tanto iremos utilizar o alinhamento do tipo parágrafo para definir esta coluna e definiremos a largura desejada desta coluna.

Listagem 6.1: Exemplo de tabela com parágrafo em uma das colunas.

[1] Os três modos de alinhamento de parágrafo para as colunas de uma tabela se referem ao alinha vertical. São eles: `p` para alinhamento no topo (este é o modo usual), `m` para alinhamento no meio e `b` para alinhamento na parte inferiro (os dois últimos modos requerem a utilização do pacote `array`).

```latex
\begin{tabular}{|l|p{8cm}|}
\hline
dia & mensagem da sorte \\
\hline\hline
2a & ``Wagner's music is better than it sounds.'' (Mark Twain) \\ \hline
3a & ``It usually takes more than three weeks to prepare a good impromptu speech.'' (Mark Twain) \\ \hline
4a & ``I've touch'd the highest point of all my greatness; \newline
And from that full meridian of my glory\newline
I haste now to my setting.  I shall fall,\newline
Like a bright exhalation in the evening\newline
And no man see me more.'' (Shakespeare)
\\ \hline
\end{tabular}
```

dia	mensagem da sorte
2a	"Wagner's music is better than it sounds." (Mark Twain)
3a	"It usually takes more than three weeks to prepare a good impromptu speech." (Mark Twain)
4a	"I've touch'd the highest point of all my greatness; And from that full meridian of my glory I haste now to my setting. I shall fall, Like a bright exhalation in the evening And no man see me more." (Shakespeare)

Caso nos deparemos com uma tabela contendo múltiplas colunas, podemos defini-las todas de uma só vez, devemos então expressar qual o número de colunas e qual o tipo de alinhamento que elas receberão. Veja o exemplo apresentado na Listagem 6.3. Definimos uma tabela que terá uma coluna alinhada à esquerda (`l`) e três colunas com alinhamento central (`*{3}{c}`).

Listagem 6.3: Exemplo de definição múltipla.

coordenadas	x	y	z
p1	1	0	0
p2	0	1	0
p3	0	0	1

```
\begin{tabular}{l|*{3}{c}}
\hline
coordenadas    & x & y & z \\ \hline
p1             & 1 & 0 & 0 \\
p2             & 0 & 1 & 0 \\
p3             & 0 & 0 & 1 \\ \hline
\end{tabular}
```

Outro recurso recorrente é a expansão de células ao longo das colunas, formando uma célula maior. Deveremos então informar quantas colunas serão agregadas e qual será o novo alinhamento dado. Vamos reescrever o exemplo anterior utilizando esta opção. Na primeira linha da tabela utilizaremos o comando `\multicolumn` para realizar a expansão de uma célula. No exemplo dado na Listagem 6.4, será feita uma expansão de forma que a célula final ocupará 3 colunas e terá alinhamento central. Note que entre a segunda e terceira linhas, ao invés de utilizar o `\hline` estamos utilizando o `\cline` para criar uma linha horizontal parcial, cobrindo as colunas referenciadas como argumento.

Listagem 6.4: Utilização de múltiplas colunas.

	coordenadas		
	x	y	z
p1	1	0	0
p2	0	1	0
p3	0	0	1

```
\begin{tabular}{l|*{3}{c}}
\hline
 & \multicolumn{3}{c}{coordenadas} \\
\cline{2-4}
   & x & y & z \\ \hline
p1 & 1 & 0 & 0 \\
p2 & 0 & 1 & 0 \\
p3 & 0 & 0 & 1 \\ \hline
\end{tabular}
```

Outra forma de expandir as células através de linhas. Para tanto é necessário utilizar o pacote `multirow`. O exemplo da Listagem 6.5 ilustra a utilização do comando `\multirow` para mesclar linhas.

Listagem 6.5: Mesclando células ao longo de linhas.

	x	y	z
mescladas	1	0	0
	0	1	0
	0	0	1

```
\begin{tabular}{l|*{3}{c}}
\hline
   & x & y & z \\ \hline
\multirow{3}{5em}{mescladas} & 1 & 0
    & 0 \\
 & 0 & 1 & 0 \\
 & 0 & 0 & 1 \\ \hline
\end{tabular}
```

As tabelas podem receber legendas e rótulos para serem referenciadas. Para tando deveremos utilizar o ambiente `table`.

Listagem 6.6: Utilização do ambiente `table`.

Neste exemplo mostramos a Tabela 6.1.

Tabela 6.1: Tabela Exemplo

1	2	3
4	5	6
7	8	9

```
Neste exemplo mostramos a \Cref{tbl
    01}.

\begin{table}[h!]
\centering
\caption{Tabela Exemplo}
\label{tbl01}
\begin{tabular}{ccc}
1 & 2 & 3 \\
4 & 5 & 6 \\
7 & 8 & 9 \\
\end{tabular}

\end{table}
```

6.3 Tabelas longas

Algumas vezes queremos mostrar muitos dados compilados em um única tabela, que pode precisar se estender por várias páginas. Iremos mostrar aqui a junção de duas ferramentas. Podemos usar o pacote `longtable` para criar tabelas longas. O seu funcionamento é bem semelhante ao `tabular`. Mas é muito inconveniente ter que colocar todos os dados de uma longa tabela em nosso arquivo `.tex`. Além disso, usualmente esses dados são provenientes de alguma planilha ou gerados por algum programa. Podemos carregar esses dados facilmente se utilizarmos o pacote `csvsimple`. Abaixo apresentamos um código simples para carregar os dados do arquivo `binary.csv` e inseri-los em uma longa tabela. Não iremos apresentar os resultados aqui para

economizar espaço. O arquivo `binary.csv` contém duas colunas. A primeira linha do arquivo é o cabeçalho, onde é fornecido o nome das colunas, nome este que será utilizado pelo `csvreader` para preencher os dados na tabela. No exemplo em questão, uma coluna possui o nome `numero` e a outra coluna possui o nome `binário`.

Listagem 6.2: Código em `bash` para gerar os dados para a tabela com a representação decimal e binária de um número.

```
$ (echo "numero,binario"; for ((i=0; i<=255; i++)); do echo -ne
↪  "$i,"; echo "obase=2;$i" | bc; done) > Tabelas/binary.csv
```

Listagem 6.3: Exemplo de tabela longa.

```
\begin{longtable}[c]{l|l}%
\bfseries numero & \bfseries binário
\csvreader[head to column names]{Tabelas/binary.csv}{}
{\\\hline\numero & \binario}
\end{longtable}
```

Outro pacote capaz de criar tabelas longas é o `supertabular`. Caso seu documento tenha duas colunas e você deseja que a tabela se extenda ao longo das colunas do texto, deverá utilizar este pacote. Apresentamos na Listagem 6.4 um exemplo de utilização deste. O resultado pode ser visto na Figura 6.2.

Listagem 6.4: Exemplo de tabela longa utilizando o `supertabular`.

```
\begin{supertabular}[c]{l|l}%
\bfseries numero & \bfseries binário
\csvreader[head to column names]{binary.csv}{}
{\\\hline\numero & \binario}
\\
\end{supertabular}
```

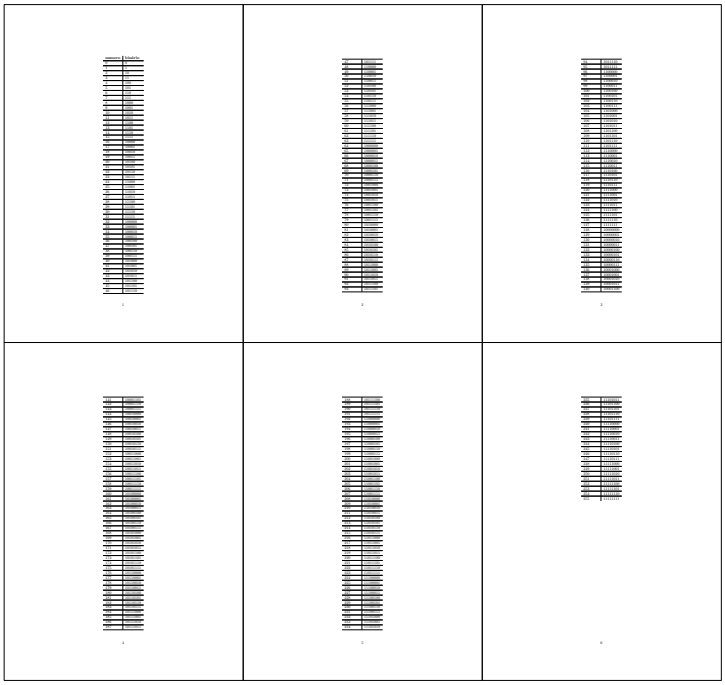
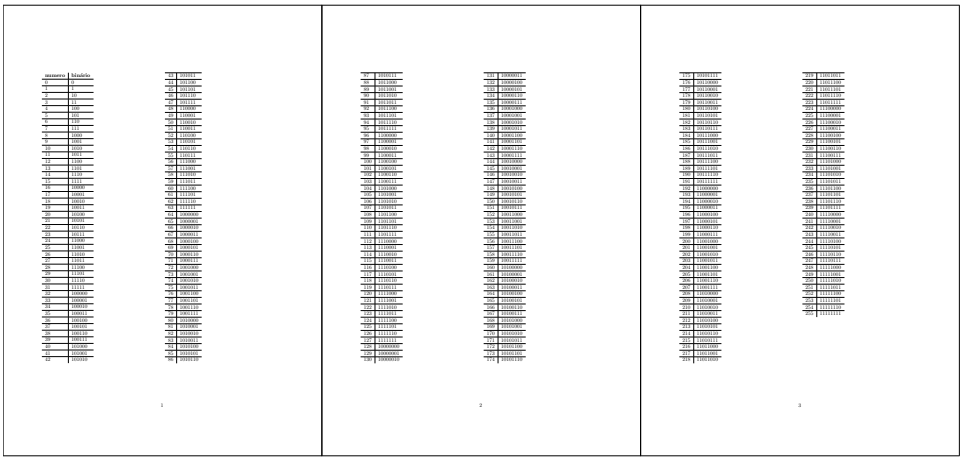


Figura 6.1: Exemplo de tabela longa gerada pela Listagem 6.3.



numero	binário
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001
26	11010
27	11011
28	11100
29	11101
30	11110
31	11111
32	100000
33	100001
34	100010
35	100011
36	100100
37	100101
38	100110
39	100111
40	101000
41	101001
42	101010
43	101011
44	101100
45	101101
46	101110
47	101111
48	110000
49	110001
50	110010
51	110011
52	110100
53	110101
54	110110
55	110111
56	111000
57	111001
58	111010
59	111011
60	111100
61	111101
62	111110
63	111111
64	1000000
65	1000001
66	1000010
67	1000011
68	1000100
69	1000101
70	1000110
71	1000111
72	1001000
73	1001001
74	1001010
75	1001011
76	1001100
77	1001101
78	1001110
79	1001111
80	1010000
81	1010001
82	1010010
83	1010011
84	1010100
85	1010101
86	1010110
87	1010111
88	1011000
89	1011001
90	1011010
91	1011011
92	1011100
93	1011101
94	1011110
95	1011111
96	1100000
97	1100001
98	1100010
99	1100011
100	1100100
101	1100101
102	1100110
103	1100111
104	1101000
105	1101001
106	1101010
107	1101011
108	1101100
109	1101101
110	1101110
111	1101111
112	1110000
113	1110001
114	1110010
115	1110011
116	1110100
117	1110101
118	1110110
119	1110111
120	1111000
121	1111001
122	1111010
123	1111011
124	1111100
125	1111101
126	1111110
127	1111111
128	10000000
129	10000001
130	10000010
131	10000011
132	10000100
133	10000101
134	10000110
135	10000111
136	10001000
137	10001001
138	10001010
139	10001011
140	10001100
141	10001101
142	10001110
143	10001111
144	10010000
145	10010001
146	10010010
147	10010011
148	10010100
149	10010101
150	10010110
151	10010111
152	10011000
153	10011001
154	10011010
155	10011011
156	10011100
157	10011101
158	10011110
159	10011111
160	10100000
161	10100001
162	10100010
163	10100011
164	10100100
165	10100101
166	10100110
167	10100111
168	10101000
169	10101001
170	10101010
171	10101011
172	10101100
173	10101101
174	10101110
175	10101111
176	10110000
177	10110001
178	10110010
179	10110011
180	10110100
181	10110101
182	10110110
183	10110111
184	10111000
185	10111001
186	10111010
187	10111011
188	10111100
189	10111101
190	10111110
191	10111111
192	11000000
193	11000001
194	11000010
195	11000011
196	11000100
197	11000101
198	11000110
199	11000111
200	11001000
201	11001001
202	11001010
203	11001011
204	11001100
205	11001101
206	11001110
207	11001111
208	11010000
209	11010001
210	11010010
211	11010011
212	11010100
213	11010101
214	11010110
215	11010111
216	11011000
217	11011001
218	11011010
219	11011011
220	11011100
221	11011101
222	11011110
223	11011111
224	11100000
225	11100001
226	11100010
227	11100011
228	11100100
229	11100101
230	11100110
231	11100111
232	11101000
233	11101001
234	11101010
235	11101011
236	11101100
237	11101101
238	11101110
239	11101111
240	11110000
241	11110001
242	11110010
243	11110011
244	11110100
245	11110101
246	11110110
247	11110111
248	11111000
249	11111001
250	11111010
251	11111011
252	11111100
253	11111101
254	11111110
255	11111111

1 2 3

Figura 6.2: Exemplo de tabela longa gerada pela Listagem 6.4.

Capítulo 7

Listas

Sumário

7.1 Introdução 90
7.2 Listas não numeradas simples 90
7.3 Listas não numeradas aninhadas 91
7.4 Listas numeradas 93
7.5 Listas com descrição 99
7.6 Listas aninhadas 100
7.7 Listas com várias colunas 101

7.1 Introdução

Listas são estruturas importantes em um texto para destacar informações.A estrutura e formatação da listas faz com que sejam facilmente notadas na estrutura de uma página. As listas também facilitam a leitura e a organização do texto. Por exemplo, quando temos uma sequência de instruções sequenciais, a estruturação do texto de forma que cada passo esteja em um item da lista facilita a leitura e o entendimento do processo.

Existem diferentes tipos de listas e cada uma delas deve ser utilizada segundo o seu propósito, facilitando o entendimento, a memorização e o posterior resgate dos pontos importantes destacados na lista.

A formatação da lista é importante. Deve-se controlar sua indentação, espaçamento entre itens e espaçamento entre as linhas dentro de um mesmo item. Estes parâmetros são controlados automaticamente pelo LaTeX e em casos específicos podemos ajustá-los para atender a algum requisito.

De forma geral, as listas podem ser numeradas ou não numeradas, estruturadas verticalmente ou em sentença. Preferencialmente, devemos utilizar listas numeradas para elencar itens que requerem uma ordem (como os passo de uma receita) e listas não numeradas para itens que não requerem uma ordem. Usamos listas em sentença quando não queremos destacar a lista do texto, deixando-a mais incorporada ao texto. Quando queremos dar mais ênfase à lista, utilizamos a lista organizada verticalmente.

7.2 Listas não numeradas simples

O ambiente `itemize` permite criar listas sem numeração. Para cada item da lista deve-se usar o comando `item`. Como primeiro exemplo vamos criar uma lista simples como mostrado a seguir:

- arroz
- suco
- frutas

Para produzir a lista anterior utilizamos o código da Listagem 7.1. Perceba que cada elemento da lista deve ser precedido de um comando `item`.

Listagem 7.1: Listas não numeradas simples.

```
\begin{itemize}
  \item arroz
  \item suco
```

```
  \item frutas
\end{itemize}
```

Se quiser usar outro tipo de símbolo na sua lista basta mudar a opção de símbolo através do uso do parâmetro `label`:

- arroz
- suco
- frutas

Neste caso utilize o pacote `enumitem`. A Listagem 7.2 apresenta o código para mudança do símbolo da lista. Observe que inserimos o comando `label={--}` como opção do ambiente.

Listagem 7.2: Alterando o símbolo da lista não numerada simples.

```
\begin{itemize}[label={--}]
  \item arroz
  \item suco
  \item frutas
\end{itemize}
```

7.3 Listas não numeradas aninhadas

Para criar listas aninhadas basta criar um ambiente `itemize` dentro de um `item` de um ambiente `itemize`. Como exemplo, vamos criar uma lista de compras da seguinte forma:

- Materiais de Limpeza
  - Detergente
  - Sabão
    - Sabão em Pó
    - Sabão em barra
- Higiene Pessoal
  - Sabonete
  - Pasta de dente
- Açougue
  - Frango

  - Carne de Boi

Observe que o símbolos para marcar o nível do item na lista varia. A lista mais externa recebe um símbolo e a mais interna outro. O LaTeX faz isso automaticamente, mas se necessário você pode especificar cada um deles, conforme vimos na Listagem 7.2.

Para isto vamos utilizar a estrutura da Listagem 7.3. Observe que temos um ambiente `itemize` dentro de outro ambiente `itemize`. No código utilizamos a indentação para facilitar a visualização do aninhamento das listas.

Listagem 7.3: Listas numeradas aninhadas simples.

```
\begin{itemize}
  \item Materiais de Limpeza
  \begin{itemize}
    \item Detergente
    \item Sabão
    \begin{itemize}
      \item Sabão em Pó
      \item Sabão em barra
    \end{itemize}
  \end{itemize}
  \item Higiene Pessoal
  \begin{itemize}
    \item Sabonete
    \item Pasta de dente
  \end{itemize}
  \item Açougue
  \begin{itemize}
    \item Frango
    \item Carne de Boi
  \end{itemize}
\end{itemize}
```

Quando você precisar de mais níveis em sua lista é preciso usar mais estruturas de `itemize` para a sua lista. Dependendo do número de níveis, você poderá receber um erro `LaTeX Error: Too deeply nested` que significa que atingiu o número máximo de níveis permitidos. Como exemplo vamos criar uma lista com 3 níveis como apresentado a seguir:

- Nação
  - Estado
    - Cidade

O código é apresentada na Listagem 7.4.

Listagem 7.4: Listas não numeradas simples com 3 níveis.

```
\begin{itemize}
  \item Nação
  \begin{itemize}
    \item Estado
    \begin{itemize}
      \item Cidade
    \end{itemize}
  \end{itemize}
\end{itemize}
```

Podemos criar listas aninhadas customizadas, para isto vamos alterar nosso exemplo para mudar os símbolos de cada um dos níveis de nossa lista.

* Nação
  - Estado
    . Cidade

O código para customização dos símbolos de cada nível é apresentada na Listagem 7.5.

Listagem 7.5: Customização de símbolos em listas não numeradas simples com 3 níveis.

```
\begin{itemize}[label={*}]
  \item Nação
  \begin{itemize}[label={--}]
    \item Estado
    \begin{itemize}[label={.}]
      \item Cidade
    \end{itemize}
  \end{itemize}
\end{itemize}
```

7.4 Listas numeradas

O ambiente `enumerate` permite criar listas com numeração como apresentado a seguir:

1. arroz
2. feijão

3. óleo

A Listagem 7.6 apresenta o código do exemplo anterior. Neste caso trocamos o ambiente `itemize` pelo `enumerate`.

Listagem 7.6: Listas Numeradas Simples

```
\begin{enumerate}
  \item arroz
  \item feijão
  \item óleo
\end{enumerate}
```

Você também pode criar listas numeradas e aninhadas como no exemplo a seguir.

1. Frutas
   (a) Laranja
2. Higiene
   (a) Sabonete

Neste caso utilizamos `enumerate` em todos os níveis da lista. A Listagem 7.7 apresenta o código para o exemplo anterior.

Listagem 7.7: Listas Numeradas e Aninhadas

```
\begin{enumerate}
  \item Frutas
  \begin{enumerate}
    \item Laranja
  \end{enumerate}
  \item Higiene
  \begin{enumerate}
    \item Sabonete
  \end{enumerate}
\end{enumerate}
```

Caso queira utilizar números romanos para enumerar sua lista utilize o pacote `enumitem`. Este pacote deve ser carregado no preâmbulo do documento LaTeX através do comando da Listagem 7.8.

Listagem 7.8: Pacote para Inserir Pacotes para Listas Numeradas em Algarismos Romanos

```
\usepackage{enumitem}
```

Para criar listas numeradas com algarismos romanos iremos utilizar o ambiente **enumerate** e como parâmetro iremos especificar o rótulo dos itens, como pode ser visto na Listagem 7.9.

(i) arroz

(ii) cerveja

(iii) óleo

(iv) vinho

(v) carne

(vi) vinage

(vii) pão

(viii) temperos

(ix) sal

Listagem 7.9: Exemplo de listas numeradas em algarismos romanos.

```
\begin{enumerate}[label=(\roman*)]
  \item arroz
  \item cerveja
  \item óleo
  \item vinho
  \item carne
  \item vinagre
  \item pão
  \item temperos
  \item sal
\end{enumerate}
```

É possível também criar listas numeradas com algarismos romanos maiúsculos como no exemplo a seguir.

(I) arroz

(II) cerveja

(III) óleo

(IV) vinho

(V) vinagre

Para gerar esta lista utilizamos o código apresentado na Listagem 7.10.

Listagem 7.10: Exemplo de listas numeradas em algarismos romanos maiúsculos.

```
\begin{enumerate}[label=(\Roman*)]
  \item arroz
  \item cerveja
  \item óleo
  \item vinho
  \item vinagre
\end{enumerate}
```

Também podemos criar listas com algarismos romanos aninhadas e alterar entre letras maiúsculas e minúsculas como no exemplo a seguir:

(I) Lista 1

  (i) arroz
  (ii) cerveja
  (iii) óleo

(II) Lista 2

  (i) arroz
  (ii) cerveja
  (iii) óleo

A Listagem 7.11 apresenta o exemplo a criação de uma lista apresentada no exemplo anterior. Para resolver reste problema utilizaremos as duas estruturas apresentadas anteriormente:

- `\begin{enumerate}[label=(\Roman*)]` para criar a lista com letras maiúsculas

- `\begin{enumerate}[label=(\roman*)]` para criar a lista com letras minúsculas.

Não se esqueça que será preciso utilizar estruturas de `enumerate` recursivas para que o sistema crie as listas aninhadas.

Listagem 7.11: Exemplo de listas aninhadas numeradas em algarismos romanos maiúsculos e minúsculos.

```
\begin{enumerate}[label=(\Roman*)]
\item Lista 1
  \begin{enumerate}[label=(\roman*)]
    \item arroz
    \item cerveja
    \item óleo
  \end{enumerate}
\item Lista 2
  \begin{enumerate}[label=(\roman*)]
    \item arroz
    \item cerveja
    \item óleo
  \end{enumerate}
\end{enumerate}
```

É possível utilizar listas com letras minúsculas e como exemplo vamos criar uma lista de compras como apresentado a seguir (apenas com letras minúsculas). Lembre-se de inserir o pacote `enumitem`.

(a) arroz

(b) óleo

(c) vinho

(d) carne

(e) vinagre

(f) pão

A Listagem 7.12 apresenta o código da lista anterior.

Listagem 7.12: Exemplo de listas com letras minúsculas.

```latex
\begin{enumerate}[label=(\alph*)]
  \item arroz
  \item óleo
  \item vinho
  \item carne
  \item vinagre
  \item pão
\end{enumerate}
```

É possível também utilizar enumerações com letras maiúsculas, assim sua lista utilizará as letras A, B, C, etc. Vamos criar uma lista de compras como apresentado a seguir (apenas com letras maiúsculas). Lembre-se de inserir o pacote `enumitem`.

(A) arroz

(B) óleo

(C) vinho

(D) carne

(E) vinagre

(F) pão

(G) torradas

(H) uvas

A Listagem 7.13 apresenta o código da lista anterior.

Listagem 7.13: Exemplo de Listas com Letras Maiúsculas

```latex
\begin{enumerate}[label=(\Alph*)]
  \item arroz
  \item óleo
  \item vinho
  \item carne
  \item vinagre
  \item pão
  \item uvas
\end{enumerate}
\end{Verbatim}
```

Para utilizar letras maiúsculas utilizamos `label=(\Alph*)` e para letras minúsculas `label=(\alph*)`.

Vamos mesclar os dois exemplos das Listagens 7.12 e 7.13 e produzir listas aninhadas.

(A) Lista 1

  (a) arroz
  (b) óleo

(B) Lista 2

  (a) carne
  (b) vinagre

A Listagem 7.14 apresenta o resultado do exemplo anterior.

Listagem 7.14: Exemplo de listas aninhadas com letras maiúsculas e minúsculas.

```
\begin{enumerate}[label=(\Alph*)]
  \item Lista 1
  \begin{enumerate}[label=(\alph*)]
    \item arroz
    \item óleo
  \end{enumerate}
  \item Lista 2
  \begin{enumerate}[label=(\alph*)]
    \item carne
    \item vinagre
  \end{enumerate}
\end{enumerate}
```

7.5 Listas com descrição

Para fazer listas com elementos de descrição utilize o ambiente `description`. A lista a seguir apresenta um exemplo de sua utilização.

Raças de Cachorros: labrador, beagle, boxer, etc ...

Raças de Gatos: Angorá, Persa, etc ...

A Listagem 7.15 apresenta o código para inserir listas com descrição.

Listagem 7.15: Exemplo de listas aninhadas com descrição.

```
\begin{description}
  \item[Raças de Cachorros] labrador, beagle, boxer, etc...
  \item[Raças de Gatos] Angorá, Persa, etc..
\end{description}
```

Se quiser colocar a descrição em outra linha como a seguir utilize o código da Listagem 7.16.

Raças de Cachorros
labrador, beagle, boxer, etc ...

Raças de Gatos
Angorá, Persa, etc ...

Listagem 7.16: Exemplo de listas aninhadas com descrição.

```
\begin{description}
  \item[Raças de Cachorros] \hfill \\ labrador, beagle, boxer, etc \ldots
  \item[Raças de Gatos] \hfill \\  Angorá, Persa, etc \ldots
\end{description}
```

Observe que para criar os três pontos após etc utilizamos o comando `\ldots`. Outra diferença é o uso de `description` para definir que a lista será do tipo de descrição.

7.6 Listas aninhadas

É possível criar listas aninhadas utilizando o LaTeX, como já vimos nos exemplos anteriores. Podemos inclusive misturar os tipos de listas que estarão aninhadas. Cada nível deve ser iniciado com `begin` e `end`. A lista a seguir apresenta o conceito de listas aninhadas.

Nível 1 Lista não numerada:

- Nível 1.1
  - Nível 1.1.1
  - Nível 1.1.2

Nível 2 Lista numerada:

1. Nível 2.1
   (a) Nível 2.1.1
   (b) Nível 2.1.2
2. Nível 2.2

A Listagem 7.17 apresenta o código que gera a lista anterior.

Listagem 7.17: Exemplo de listas aninhadas com diferentes tipos.

```
\begin{description}
  \item[Nível 1] Lista não numerada:
    \begin{itemize}
      \item Nível 1.1
      \begin{itemize}
        \item Nível 1.1.1
        \item Nível 1.1.2
      \end{itemize}
    \end{itemize}
  \item[Nível 2] Lista numerada:
    \begin{enumerate}
      \item Nível 2.1
      \begin{enumerate}
        \item Nível 2.1.1
        \item Nível 2.1.2
      \end{enumerate}
      \item Nível 2.2
    \end{enumerate}
\end{description}
```

7.7 Listas com várias colunas

Para expandir listas ao longo de várias colunas é necessário incluir o pacote `multicol`. Pode-se criar com um número customizado de colunas.

Exemplos com três colunas:

- elemento 1
- elemento 2
- elemento 3
- elemento 4
- elemento 5
- elemento 6

A Listagem 7.18 apresenta o exemplo para 3 colunas.

Listagem 7.18: Exemplo de Lista com 3 Colunas

```latex
\begin{multicols}{3}
  \begin{itemize}
    \item elemento 1
    \item elemento 2
    \item elemento 3
    \item elemento 4
    \item elemento 5
    \item elemento 6
  \end{itemize}
\end{multicols}
```

Capítulo 8

Fórmulas Matemáticas

Sumário

8.1 Formalismo matemático 104

8.2 Fórmulas matemáticas no LaTeX 104

8.2.1 Operadores 107

8.2.2 Parênteses, colchetes e chaves 110

8.2.3 Somatórios, integrais e produtórios 110

8.2.4 Matrizes 111

8.3 Teoremas e definições 113

8.4 Formatação de caracteres 114

8.5 Diacríticos 115

8.6 Lista de símbolos 116

8.1 Formalismo matemático

Matemática requer formalismo, e este formalismo deve ser feito com precisão. As equações matemáticas devem ser capazes de abstrair os conceitos a partir de objetos do mundo real, onde originalmente estão conectados, e assim generalizar para que uma determinada formulação seja capaz de abarcar uma grande quantidade de fenômenos e descrever problemas semelhantes.

A notação matemática deve utilizar-se de símbolos para denotarem representações bem definidas de quantidades e operadores que representam alguma manipulação sobre essas quantidades. Ao longo da história da matemática, a notação hoje utilizada foi estabelecida gradualmente, partindo de uma forma de retórica pura, até chegar à notação simbólica que hoje aprendemos em sala de aula.

Ao criar o TeX, Donald Knuth preocupou-se em estabelecer uma forma simples e generalista para descrever as formulações matemáticas atuais através de uma representação por caracteres ASCII. As equações deveriam ter uma apresentação profissional quando impressas e deveriam ser sempre a mesma apresentação em qualquer computador, ou qualquer sistema. Talvez esta característica tenha sido o diferencial para levar o TeX e LaTeX a serem muito utilizados no meio acadêmico-científico, sobretudo nas áreas de ciências exatas e engenharias. A notação matemática criada por Knuth é hoje também utilizada na Wikipedia e muitas outras ferramentas, inclusive é uma das formas possíveis de se inserir equações no Microsoft Word.

Além de se preocupar com a representação e usabilidade, Knuth também tinha como uma de suas principais preocupações a apresentação. Destarte, baseou-se nos trabalhos que julgava apresentar as formalismo matemático em sua forma mais bela para criar a sua tipografia para os equações matemáticas no TeX[4, 5].

8.2 Fórmulas matemáticas no LaTeX

Um documento simples, em que se utilizará apenas algumas poucas equações, sem sofisticação, poderá ser redigito sem o auxílio de nenhum pacote extra, pois o LaTeX já possui as ferramentas básicas necessárias, conforme vimos no primeiro exemplo, apresentado na Seção 1.1. Entretanto, para os usuários das áreas científicas, será comum a utilização de formulas mais complexas, fazendo-se necessária a utilização do pacote `amsmath`.

Para escrever documentos com equações mais complexas, deve-se utilizar os pacotes `mathtools` ou `amsmath`. O pacote `mathtools` automaticamente

carrega o pacote `amsmath` e ainda fornece mais algumas funcionalidades e altera alguns funcionamentos peculiares do `amsmath`. Devemos incluir no preambulo do documento uma das seguintes linhas para carregar o pacote:

Listagem 8.1: Pacotes matemáticos.

```
\usepackage{amsmath}
% ou
\usepackage{mathtools}
```

Em um texto, para inserir equações, devemos utilizar o ambiente matemático, isto é feito inserindo a equação entre os marcadores do ambiente: `\[ · \]`, `\( · \)`, `$ · $` ou ainda utilizando o ambiente `equation`.

Listagem 8.2: Ambiente de equações.

```
\begin{equation}
equacao
\end{equation}
```

Na Listagem 8.1 apresentamos alguns exemplos de utilização.

Listagem 8.1: Exemplos de utilização das equações.

Seja x a variavel independente e y a variavel dependente, sabemos que existe a seguinte relacao entre elas:

$$y = ax + b, \tag{8.1}$$

onde a e b sao dois parametros. Conhecendo os valores

$$a = 1$$

e

$$b = -1,$$

na equacao 8.1, poderemos calcular o valor de y. Quando $x = 2$ teremos $y = 1$.

```
Seja $x$ a variavel
    independente e $y$ a
    variavel dependente,
sabemos que existe a seguinte
    relacao entre elas:
\begin{equation}\label{eq-ex-
    parabola}
y = ax+b ,
\end{equation}
onde $a$ e $b$ sao dois
    parametros. Conhecendo os
    valores
\[a=1\] e \[b=-1 ,\] na equacao
     \ref{eq-ex-parabola},
poderemos calcular o valor de $
    y$. Quando \(x=2\) teremos
    $y=1$.
```

Note que no exemplo acima que todas as formas de definição de equação foram utilizadas. Quando utilizamos `$`, estamos definindo uma equação que aparecerá na linha do texto. As demais formas criaram equações que ficarão destacadas do texto. Ao declarar a equação da parábola, utilizamos o ambiente `equation`, sendo assim possível dar um rótulo a ela (utilizando o `\label`), sendo assim possível referenciá-la posteriormente, utilizando o `\ref` (como apresentado no exemplo).

Como podemos constatar nos primeiros exemplos, a utilização de equações na LaTeX é bem simples, mas é necessário conhecer alguns comandos para utilizar de forma prática e rápida. É necessário que se tenha prática, então utilize o LaTeX sempre que possível, pois apenas assim ganha-se familiaridade para se fazer um bom uso de todos os seu recursos.

Sempre que necessário você deverá recorrer a este livro ou rapidamente buscar na internet para descobrir como se deve fazer para escrever uma determinada equação que se tem em mente. Ao final deste capítulo será apresentada uma secção com uma lista de símbolos (originalmente compilada por David Carlisle) que poderá ser consultada sempre que necessário.

Frações e binômios

Para construir frações podemos utilizar os comandos `\frac{·}{·}`, `\dfrac{·}{·}` ou `\tfrac{·}{·}`. A Equação (8.2) apresenta o uso do `\frac` e a Equação (8.3) apresenta o uso de `\dfrac` e `\tfrac`.

$$a = \frac{1}{5} \tag{8.2}$$

$$a = \frac{1}{5} = \tfrac{1}{5} \tag{8.3}$$

A Listagem 8.3 apresenta o código para gerar os exemplos das Equações 8.2 e 8.3.

Listagem 8.3: Frações em LaTeX

```
# uso do \frac{·}{·}
\begin{equation} \label{frac}
a = \frac{1}{5}
\end{equation}

# uso do \dfrac{·}{·}
\begin{equation} \label{dfrac}
a = \dfrac{1}{5} = \tfrac{1}{5}
\end{equation}
```

A diferença entre os comandos está no modo de exibição. `\dfrac` e `\tfrac` são definidos no pacote `amsmath`, o primeiro utiliza o modo display (d) e o segundo o modo texto (t). Já o comando `\frac` escolhe automaticamente o melhor modo de acordo com o contexto. O mesmo código que gerou a Equação (8.3), quando inserido na equação dentro do texto, gera o seguinte resultado: $a = \frac{1}{5} = \dfrac{1}{5} = \tfrac{1}{5}$.

Quando queremos que a fração apareça no meio do texto, recorremos a outros comandos (ou artifícios) para apresentá-las de forma mais elegante. Vejamos o exemplo abaixo onde ilustramos algumas formas de apresentar as frações.

Listagem 8.2: Exemplos de utilização de frações e binômios

O coeficiente binomial e definido por

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{8.4}$$

Para representar o valor 0.5 em forma fracionaria, utilizamos $\frac{1}{2}$ (ou podermos apresenta-lo como $^1/_2$). Podemos ainda utilizar o pacote `xfrac` e assim apresentar na seguinte forma: $\sfrac{1}{2}$.

```
O coeficiente binomial e
    definido por
\begin{equation}
\binom{n}{k} = \frac{n!}{k!(n-k
    )!}
\end{equation}
Para representar o valor $0.5$
    em forma fracionaria,
utilizamos $\tfrac{1}{2}$ (ou
    podermos apresenta-lo como
    $^1/_2$).
Podemos ainda utilizar o pacote
     \texttt{xfrac} e assim
    apresentar
na seguinte forma: $\sfrac
    {1}{2}$.
```

8.2.1 Operadores

Os operadores, funções trigonométricas, logaritmo, exponencial, dentre outras, aparecem nas equações escritos como palavras, entretanto não devemos simplesmente escrever a palavra desejada pois a formatação utilizada para apresentar uma palavra, representando um operador, e uma sequência de caracteres qualquer é diferentes. Por exemplo, para expressar o cosseno de theta, devemos utilizar $\cos\theta$ ao invés de $cos\theta$. Nesta caso, utilizamos `\cos` ao invés de simplesmente `cos`.

Além disso, alguns operadores podem receber parâmetros que serão tratados de maneira diferente e possuem comportamento diferente dependendo de como são utilizados. A Listagem 8.3 apresenta alguns exemplos.

Listagem 8.3: Exemplos de operadores.

$$\lim_{x \rightarrow h} (x-h) \tag{8.5}$$

Tomando o limite $\lim_{x \rightarrow h} (x-h)$ teremos...

$$\sum_{n=0}^{N} x_n \tag{8.6}$$

a soma dos $N+1$ termos $\sum_{n=0}^{N} x_n$ ...

```
\begin{equation}
\lim_{x \rightarrow h} (x-h)
\end{equation}

Tomando o limite $\lim_{x \
    rightarrow h} (x-h)$
    teremos...

\begin{equation}
\sum_{n=0}^{N} x_n
\end{equation}

a soma dos $N+1$ termos $\sum_{
    n=0}^{N} x_n$ ...
```

Listagem 8.4: Exemplos do operador para a função cosseno

```
Na equação abaixo utilizaremos a simples sequencia de caracteres `cos'.
\begin{equation}
f(x) = cos(x)
\end{equation}
Observe como o resultado visual eh bem diferente daquele quando
utilizamos a forma correta, ou seja, o operador.
\begin{equation}
f(x) = \cos(x)
\end{equation}
```

Listagem 8.4: Exemplos do operador para a função cosseno.

Na equacao abaixo utilizaremos a simples sequencia de caracteres ‘cos’.

$$f(x) = cos(x) \tag{8.7}$$

Observe como o resultado visual eh bem diferente daquele quando utilizamos a forma correta, ou seja, o operador.

$$f(x) = \cos(x) \tag{8.8}$$

```
Na equacao abaixo utilizaremos
    a simples sequencia de
    caracteres `cos'.
\begin{equation}
f(x) = cos(x)
\end{equation}
Observe como o resultado visual
     eh bem diferente daquele
    quando utilizamos a forma
correta, ou seja, o operador.
\begin{equation}
f(x) = \cos(x)
\end{equation}
```

Abaixo apresentamos alguns exemplos de utilização dos operadores.

Listagem 8.5: Exemplos de operadores.

$$f(x) = \sin^2(x) - \arctan(x) \tag{8.9}$$

$$A = \lim_{n \rightarrow \infty} \exp(a x_n) \tag{8.10}$$

$$((b))_n \triangleq a \bmod n \tag{8.11}$$

```
\begin{equation}
f(x) = \sin^2(x) - \arctan(x)
\end{equation}
\begin{equation}
A = \lim_{n \rightarrow \infty}
     \exp(a x_n)
\end{equation}
\begin{equation}
((b))_n \triangleq a \bmod n
\end{equation}
```

Definindo um novo operador

Suponha que você queira utilizar um operador matemático que não está definido no LaTeX, como por exemplo, um operador para a função seno. Neste caso, será necessário defini-lo no preambulo do documento, antes do `\begin{document}`. Vejamos um exemplo.

Listagem 8.5: Exemplo de definição de um novo operador.

```
1 \documentclass{article}
2 \usepackage{amsmath}
3 \DeclareMathOperator{\sen}{sen}
4 \begin{document}
```

```
Exemplo: $\sen(\pi)=0$.
\end{document}
```

8.2.2 Parênteses, colchetes e chaves

Parênteses, colchetes e chaves são sinais de pontuação matemática utilizados para agrupar ou separar determinados trechos de uma expressão matemática. Podemos utilizá-los em LaTeX simplesmente utilizando os caracteres normais ou, quando lidamos com expressões maiores, principalmente aquelas que envolvem frações, devemos optar por utilizá-los precedidos por `\left` ou `\right`, dependendo se estamos abrindo ou fechando o sinal de pontuação.

Listagem 8.6: Exemplos de sinais de pontuação

$$y = 2(x-1), \quad x \in \{1,2,\ldots\} \tag{8.12}$$

$$y = \left(\frac{x}{2} - 1\right) \tag{8.13}$$

```
\begin{equation}
y = 2 (x-1), \quad x \in
    \{1,2,\ldots\}
\end{equation}
\begin{equation}
y = \left( \frac{x}{2} - 1 \
    right)
\end{equation}
```

Lembre-se de que as chaves são utilizadas pelo LaTeX para definir ambientes e parâmetros de comandos. Por tanto, para utilizar chaves em um equação, devemos precedê-las da contrabarra.

8.2.3 Somatórios, integrais e produtórios

Somatórios, integrais e produtórios possuem comandos específicos. Usualmente expressamos o ponto inicial e o ponto final, ou os limites em que se são aplicados os operadores matemáticos. Nas equações em LaTeX, especificamos estes limites utilizando ˆ (para marcar o limite final) e _ (para marcar o limite final).

Listagem 8.7: Exemplos de utilização de somatórios, produtórios e integrais

$$S_n = \sum_{i=1}^n x_i \tag{8.14}$$

$$P_n = \prod_{i=1}^n x_i \tag{8.15}$$

$$I = \int_{t=0}^{\infty} e^{- t }\,\mathrm{d}t \tag{8.16}$$

$$A = \left. \frac{x^2}{2} \right|_0^1 \tag{8.17}$$

$$Q_n = \prod_{\substack{i,j\\ i\neq j}}^n x_i x_j \tag{8.18}$$

```
\begin{equation}
S_n = \sum_{i=1}^n x_i
\end{equation}
\begin{equation}
P_n = \prod_{i=1}^n x_i
\end{equation}
\begin{equation}
I = \int_{t=0}^{\infty} e^{- t
    }\,\mathrm{d}t
\end{equation}
\begin{equation}
A = \left. \frac{x^2}{2} \right
    |_0^1
\end{equation}
\begin{equation}
Q_n = \prod_{\substack{i,j\\ i
    \neq j}}^n x_i x_j
\end{equation}
```

8.2.4 Matrizes

Existe o ambiente `matrix` para definir uma matriz e vetores, e também os ambientes deste derivado, contando com diferetens formas de delimitar a matriz: `pmatrix` para utilizar parênteses, `bmatrix` para colchetes, `vmatrix` para barras verticais, dentre outros. Em todos os casos, devemos declarar elemento por elemento de uma matriz, separados por `&` e as linhas separadas por `\\`, ou seja, estrutura bem semelhante àquela utilizada para definir tabelas.

Primeiramente, veremos abaixo um exemplo bem simples de matriz 3×3. Em cada célula da matriz podemos inserir letras, números ou expressões matemáticas:

Listagem 8.8: Exemplo simples de matriz.

$$A = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & i \end{pmatrix} \tag{8.19}$$

```
\begin{equation}
A =
\begin{pmatrix}
  a & b & c \\
  d & e & f \\
  g & h & i
\end{pmatrix}
\end{equation}
```

A seguir apresentamos alguns outros exemplo um pouco mais elaborados para evidenciar formas de reproduzir matrizes em que não há necessidade de especificar todos os elementos.

Listagem 8.9: Exemplos de matrizes.

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix} \tag{8.20}$$

$$A = \begin{pmatrix} a & b & c & e & f \\ & h & i & j & k \\ & & n & o & p \\ & \mathbf{0} & & u & v \\ & & & & a \end{pmatrix} \tag{8.21}$$

```
\begin{equation}
A_{m,n} =
 \begin{pmatrix}
  a_{1,1} & a_{1,2} & \cdots &
      a_{1,n} \\
  a_{2,1} & a_{2,2} & \cdots &
      a_{2,n} \\
  \vdots & \vdots & \ddots & \
      vdots \\
  a_{m,1} & a_{m,2} & \cdots &
      a_{m,n}
 \end{pmatrix}
\end{equation}
\begin{equation}
A =
\begin{pmatrix}
a & b & c & e & f \\
  & h & i & j & k \\
  &   & n & o & p \\
  & \mbox{\Huge 0} &  & u & v
      \\
  &   &   &   & a
\end{pmatrix}
\end{equation}
```

Suponha que em um texto de estatística você queira apresentar a distribuição conjunta de um par de variáveis aleatórias X e Y sobre o alfabeto $0, 1$.

Esta matriz poderá ser criada usando o comando `bordermatrix`, conforme ilustrado na Listagem 8.10.

Listagem 8.10: Exemplo de matriz de uma distribuição conjunta.

$$p(x,y) = \begin{array}{c} x\backslash y \\ 0 \\ 1 \end{array} \begin{array}{c} \begin{array}{cc} 0 & 1 \end{array} \\ \begin{pmatrix} 0.1 & 0.2 \\ 0.2 & 0.5 \end{pmatrix} \end{array} \tag{8.22}$$

```
\begin{equation}
p(x,y) = \bordermatrix{x\
    backslash y & 0 & 1 \cr
                  0 & 0.1 & 0.2
                      \cr
                  1 & 0.2 & 0.5
                      \cr}
\end{equation}
```

8.3 Teoremas e definições

Para criar teoremas iremos utilizar o pacote `amsthm`. Desta forma, devemos adicionar a seguintes linhas ao preambulo do documento:

Listagem 8.6: Pacote para utilizar o ambiente teorema.

```
\usepackage{amsthm}
\newtheorem{theorem}{Teorema}
```

A seguir apresentamos um exemplo de utilização.

Listagem 8.11: Exemplo de teorema.

Teorema 1 (Irracionalidade de $\sqrt{2}$)**.** *A raiz quadrada de 2 e um numero irracional.*

```
\begin{theorem}[Irracionalidade
     de $\sqrt{2}$]\label{sqrt2
    thm}
A raiz quadrada de 2 e um
    numero irracional.
\end{theorem}
```

Note que utilizamos uma `label` para este teorema, de forma que poderemos referenciá-lo depois ao longo do texto.

A seguir apresentamos o exemplo com a demonstração para o teorema acima.

Listagem 8.12: Exemplo de demonstração para um teorema.

Demonstração. Suponha que $\sqrt{2}$ seja um numero racional, desta forma poderemos expressa-lo da seguinte forma:

$$\sqrt{2} = \frac{p}{q}, \tag{8.23}$$

onde p e q sao numeros inteiros sem fator comum (caso possuam, tal fator podera ser cancelado). Teremos assim, $p^2 = 2q^2$, logo p^2 e par. Para que p^2 seja par e necessario que p seja par e portanto p^2 e divisivel por 4. Assim q^2 devera ser divisivel por 2, o que so e possivel para q^2 par. Por conseguinte q tambem devera ser par e contraditoriamente p e q possuem como fator comum 2. Logo $\sqrt{2}$ devera ser irracional. □

```
\begin{proof}
Suponha que $\sqrt{2}$ seja um
    numero racional, desta
    forma poderemos expressa-lo
     da seguinte forma:
\begin{equation}
 \sqrt{2} = \frac{p}{q} ,
\end{equation}
onde $p$ e $q$ sao numeros
    inteiros sem fator comum (
    caso possuam, tal fator
    podera ser cancelado).
Teremos assim, $p^2 = 2 q^2$,
    logo $p^2$ e par. Para que
    $p^2$ seja par e necessario
     que $p$ seja par e
    portanto
$p^2$ e divisivel por $4$.
    Assim $q^2$ devera ser
    divisivel por $2$, o que so
     e possivel para $q^2$ par.
Por conseguinte $q$ tambem
    devera ser par e
    contraditoriamente $p$ e $q
    $ possuem como fator comum
    $2$.
Logo $\sqrt{2}$ devera ser
    irracional.
\end{proof}
```

8.4 Formatação de caracteres

Em algumas situações, faz-se necessário utilizar outra formatação para os caracteres em ambiente matemático, por exemplo, para denotar vetores, conjuntos, categorias, etc. No exemplo a seguir apresentamos algumas formas de formatação e exemplos de aplicação.

Listagem 8.13: Exemplo de formatação de caracteres no ambiente matemático.

$$
\begin{aligned}
\mathrm{ABCDabcd} & \quad \text{unidades de medicao}\\
\mathit{ABCDabcd} & \quad \text{nomes de variaveis}\\
\mathbf{ABCDabcd} & \quad \text{vetores e matrizes}\\
\mathcal{ABCD} & \quad \text{conjuntos e transformadas}\\
\mathbb{ABCD} & \quad \text{conjuntos especiais}
\end{aligned}
\tag{8.24}
$$

```
\begin{equation}
\begin{aligned}
\mathrm{ABCDabcd} & \quad \text
    {unidades de medicao}\\
\mathit{ABCDabcd} & \quad \text
    {nomes de variaveis}\\
\mathbf{ABCDabcd} & \quad \text
    {vetores e matrizes}\\
\mathcal{ABCD} & \quad \text{
    conjuntos e transformadas
    }\\
\mathbb{ABCD} & \quad \text{
    conjuntos especiais}
\end{aligned}
\end{equation}
```

8.5 Diacríticos

Em muitas situações desejamos usar um nome de variável para representar certo valor e queremos o mesmo nome, acrescido de um sinal gráfico complementar, para representar a mesma variável após alguma transformação, por exemplo, podemos uma certa grandeza por x e uma estimativa desta por $\hat{x}$ ou ainda $\tilde{x}$. No exemplo abaixo apresentamos alguns diacríticos utilizados para realizar tais marcações.

Listagem 8.14: Exemplo de utilização de diacríticos.

$$\begin{aligned} x' & \quad \hat{x} & \tilde{x} & \quad \dot{x} \\ \widehat{xy} & \quad \overline{xy} & \overrightarrow{xy} & \quad \vec{x} \\ \bar{x} & \quad \ddot{x} & \acute{x} & \quad \underline{x} \end{aligned} \tag{8.25}$$

```
\begin{equation}
\begin{aligned}
x'            & \quad \hat{x}
        & \quad \tilde{x}
            & \quad \dot{x} \\
\widehat{xy} & \quad \overline{
    xy} & \quad \overrightarrow
    {xy} & \quad \vec{x} \\
\bar{x}       & \quad \ddot{x}
        & \quad \acute{x}
            & \quad \underline{x
    }
\end{aligned}
\end{equation}
```

8.6 Lista de símbolos

Lista de símbolos compilada por David Carlisle para rápida consulta.

α	\alpha	θ	\theta	o	o	τ	\tau
β	\beta	ϑ	\vartheta	π	\pi	υ	\upsilon
γ	\gamma	γ	\gamma	ϖ	\varpi	ϕ	\phi
δ	\delta	κ	\kappa	ρ	\rho	φ	\varphi
ϵ	\epsilon	λ	\lambda	ϱ	\varrho	χ	\chi
ε	\varepsilon	μ	\mu	σ	\sigma	ψ	\psi
ζ	\zeta	ν	\nu	ς	\varsigma	ω	\omega
η	\eta	ξ	\xi				
Γ	\Gamma	Λ	\Lambda	Σ	\Sigma	Ψ	\Psi
Δ	\Delta	Ξ	\Xi	Υ	\Upsilon	Ω	\Omega
Θ	\Theta	Π	\Pi	Φ	\Phi		

Tabela 8.1: Letras gregas (compilada por David Carlisle)

$\pm$	\pm	$\cap$	\cap	$\diamond$	\diamond	$\oplus$	\oplus
$\mp$	\mp	$\cup$	\cup	$\triangle$	\bigtriangleup	$\ominus$	\ominus
$\times$	\times	$\uplus$	\uplus	$\bigtriangledown$	\bigtriangledown	$\otimes$	\otimes
$\div$	\div	$\sqcap$	\sqcap	$\triangleleft$	\triangleleft	$\oslash$	\oslash
$*$	\ast	$\sqcup$	\sqcup	$\triangleright$	\triangleright	$\odot$	\odot
$\star$	\star	$\vee$	\vee	$\lhd$	\lhd	$\bigcirc$	\bigcirc
$\circ$	\circ	$\wedge$	\wedge	$\rhd$	\rhd	$\dagger$	\dagger
$\bullet$	\bullet	$\setminus$	\setminus	$\unlhd$	\unlhd	$\ddagger$	\ddagger
$\cdot$	\cdot	$\wr$	\wr	$\unrhd$	\unrhd	$\amalg$	\amalg
$+$	+	$-$	-				

Tabela 8.2: Símbolos de operação binária (compilada por David Carlisle)

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

Tabela 8.3: Símbolos relacionais (compilada por David Carlisle)

,	,	;	;	:	\colon	.	\ldotp	$\cdot$	\cdotp

Tabela 8.4: Símbolos de pontuação (compilada por David Carlisle)

$\leftarrow$	\leftarrow	$\longleftarrow$	\longleftarrow	$\uparrow$	\uparrow
$\Leftarrow$	\Leftarrow	$\Longleftarrow$	\Longleftarrow	$\Uparrow$	\Uparrow
$\rightarrow$	\rightarrow	$\longrightarrow$	\longrightarrow	$\downarrow$	\downarrow
$\Rightarrow$	\Rightarrow	$\Longrightarrow$	\Longrightarrow	$\Downarrow$	\Downarrow
$\leftrightarrow$	\leftrightarrow	$\longleftrightarrow$	\longleftrightarrow	$\updownarrow$	\updownarrow
$\Leftrightarrow$	\Leftrightarrow	$\Longleftrightarrow$	\Longleftrightarrow	$\Updownarrow$	\Updownarrow
$\mapsto$	\mapsto	$\longmapsto$	\longmapsto	$\nearrow$	\nearrow
$\hookleftarrow$	\hookleftarrow	$\hookrightarrow$	\hookrightarrow	$\searrow$	\searrow
$\leftharpoonup$	\leftharpoonup	$\rightharpoonup$	\rightharpoonup	$\swarrow$	\swarrow
$\leftharpoondown$	\leftharpoondown	$\rightharpoondown$	\rightharpoondown	$\nwarrow$	\nwarrow
$\rightleftharpoons$	\rightleftharpoons	$\leadsto$	\leadsto		

Tabela 8.5: Símbolos de seta (compilada por David Carlisle)

$\ldots$	\ldots	$\cdots$	\cdots	$\vdots$	\vdots	$\ddots$	\ddots
$\aleph$	\aleph	$\prime$	\prime	$\forall$	\forall	∞	\infty
$\hbar$	\hbar	$\emptyset$	\emptyset	$\exists$	\exists	$\square$	\Box
$\imath$	\imath	∇	\nabla	$\neg$	\neg	$\diamond$	\Diamond
$\jmath$	\jmath	$\surd$	\surd	$\flat$	\flat	$\triangle$	\triangle
ℓ	\ell	$\top$	\top	$\natural$	\natural	$\clubsuit$	\clubsuit
$\wp$	\wp	$\bot$	\bot	$\sharp$	\sharp	$\diamondsuit$	\diamondsuit
$\Re$	\Re	$\\|$	\\|	$\backslash$	\backslash	$\heartsuit$	\heartsuit
$\Im$	\Im	$\angle$	\angle	∂	\partial	$\spadesuit$	\spadesuit
$\mho$	\mho	.	.	$\\|$	\\|		

Tabela 8.6: Símbolos diversos

$\sum$	\sum	$\bigcap$	\bigcap	$\bigodot$	\bigodot
$\prod$	\prod	$\bigcup$	\bigcup	$\bigotimes$	\bigotimes
$\coprod$	\coprod	$\bigsqcup$	\bigsqcup	$\bigoplus$	\bigoplus
$\int$	\int	$\bigvee$	\bigvee	$\biguplus$	\biguplus
$\oint$	\oint	$\bigwedge$	\bigwedge	$\iint$	\iint
$\iiint$	\iiint	$\iiiint$	\iiiint	$\int\cdots\int$	\idotsint

Tabela 8.7: Símbolos de tamanho variável

\arccos	\cos	\csc	\exp	\ker	\limsup	\min	\sinh
\arcsin	\cosh	\deg	\gcd	\lg	\ln	\Pr	\sup
\arctan	\cot	\det	\hom	\lim	\log	\sec	\tan
\arg	\coth	\dim	\inf	\liminf	\max	\sin	\tanh

Tabela 8.8: Símbolos do tipo logarítmo

$($	(	$)$	)	$\uparrow$	\uparrow	$\Uparrow$	\Uparrow
$[$	[	$]$	]	$\downarrow$	\downarrow	$\Downarrow$	\Downarrow
$\{$	\{	$\}$	\}	$\updownarrow$	\updownarrow	$\Updownarrow$	\Updownarrow
$\lfloor$	\lfloor	$\rfloor$	\rfloor	$\lceil$	\lceil	$\rceil$	\rceil
$\langle$	\langle	$\rangle$	\rangle	$/$	/	$\backslash$	\backslash
$\vert$	|	$\Vert$	\\|				

Tabela 8.9: Delimitadores

$\rmoustache$	\rmoustache	$\lmoustache$	\lmoustache	$\rgroup$	\rgroup	$\lgroup$	\lgroup
$\vert$	\arrowvert	$\Vert$	\Arrowvert	$\bracevert$	\bracevert		

Tabela 8.10: Delimitadores grandes

$\hat{a}$	\hat{a}	$\acute{a}$	\acute{a}	$\bar{a}$	\bar{a}	$\dot{a}$	\dot{a}	$\breve{a}$	\breve{a}
$\check{a}$	\check{a}	$\grave{a}$	\grave{a}	$\vec{a}$	\vec{a}	$\ddot{a}$	\ddot{a}	$\tilde{a}$	\tilde{a}

Tabela 8.11: Acentos no modo matemático

$\widetilde{abc}$	\widetilde{abc}	$\widehat{abc}$	\widehat{abc}
$\overleftarrow{abc}$	\overleftarrow{abc}	$\overrightarrow{abc}$	\overrightarrow{abc}
$\overline{abc}$	\overline{abc}	$\underline{abc}$	\underline{abc}
$\overbrace{abc}$	\overbrace{abc}	$\underbrace{abc}$	\underbrace{abc}
$\sqrt{abc}$	\sqrt{abc}	$\sqrt[n]{abc}$	\sqrt[n]{abc}
f'	f'	$\frac{abc}{xyz}$	\frac{abc}{xyz}

Tabela 8.12: Outras construções

Capítulo 9

Criando Índice Remissivo

Sumário

9.1 Importando o pacote 122
9.2 Inclua os índices 122
9.3 Imprima o índice 122
9.4 Compilando 123
9.5 Exemplo completo 123

9.1 Importando o pacote

Para criar um índice remissivo é necessário incluir no cabeçalho de seu documento a importação do pacote **makeidx** . Para incluir o pacote utilize o comando:

Listagem 9.1: Pacote para criar índice remissivo.

```
\usepackage{makeidx}
```

Para efetivamente fazer o índice remissivo, inclua o seguinte comando antes do início do documento:

Listagem 9.2: Comando para criar o índice remissivo.

```
\makeindex
```

Veja a utilização destes comandos na Listagem 9.4.

9.2 Inclua os índices

Para criar um novo índice utilize o comando `index` seguido do categoria e da palavra chave. Suponha que você esteja redigindo um texto e deseje criar um índice para uma determinada palavra no meio do texto. Por exemplo, ao aparecer a palavra 'corrida', deseja-se criar um índice dentro da categoria 'esportes'. Para tanto iremos utilizar o comando, conforme ilustrado na Listagem 9.3.

Listagem 9.3: Incluindo um índice.

```
Corrida\index{Esportes!Corrida}
```

Este comando foi incluído na Linha 18 da Listagem 9.4.

9.3 Imprima o índice

Antes de imprimir a bibliografia coloque o comando **printindex** como a seguir:

`\printindex`

Este comando foi incluído na Linha 20 da Listagem 9.4.

9.4 Compilando

Para compilar um documento com índice é necessário incluir o comando **makeindex** em seu editor ou no seu script de compilação. Geralmente os editores possuem em seu cabeçalho o comando para criação do índice.

9.5 Exemplo completo

A Listagem 9.4 ilustra todos os passos necessários para fazer o índice remissivo. Após compilar o documento com `latex` ou `pdflatex`, você não verá o índice criado. Para gerar o índice é necessário utilizar o processadores externos, como o `makeindex` (não se preocupe, ele vem na sua instalação do `texlive`).

Listagem 9.4: Criando Índice Remissivo.

```
1  \documentclass[11pt,a4paper]{book}
2  \usepackage[utf8]{inputenc}
3  \usepackage[portuguese]{babel}
4  \usepackage[T1]{fontenc}
5  \usepackage{makeidx}
6  \makeindex
7  \author{Alessandro Vivas Andrade}
8  \title{Exemplo}
9  \begin{document}
10 \chapter{Exemplo}
11 O Esporte Corrida \index{Esportes!Corrida}
12 \printindex
13 \end{document}
```

A Figura 9.1 apresenta o resultado do código fonte da Listagem 9.4.

Índice

Esportes
Corrida, 1

Figura 9.1: Índice Remissivo

Capítulo 10

Bibliografia

Sumário

10.1 Introdução **126**
10.2 Anatomia do documento **126**
10.3 O arquivo `.bib` **127**
10.4 Gerando a bibliografia **128**
10.5 Estilos **129**
10.6 Softwares de gerenciamento **132**

10.1 Introdução

O LaTeX permite a criação do referencial bibliográfico de maneira automática e customizada. As referências bibliográficas podem ser inseridas no próprio texto ou em um arquivo separado. Prefira trabalhar com arquivos separados, pois isto permite a utilização de softwares para gerenciar a inserção e remoção de referências. Para isto é necessário criar um arquivo com extensão `.bib` e incluir no seu documento principal.

10.2 Anatomia do documento

A Listagem 10.1 apresenta a anatomia de um documento LaTeX quando se trabalha com bibliografia. Para definir o estilo da bibliografia é necessário incluir o comando `bibliographystyle` e o comando `bibliography` que indica o nome do arquivo onde serão armazenadas as referências bibliográficas (arquivo `.bib`).

Listagem 10.1: Uso de bibliografia.

```
\documentclass[11pt,a4paper]{book}
(...)
\author{Alessando e Leonardo}
\title{Uso de Bibliografia}
\begin{document}
\maketitle
\chapter{Introdução}
Este trabalho (...)\cite{nono1878}.
(...)
\bibliographystyle{plain}
\bibliography{bibliografia}
\end{document}
```

No exemplo apresentado na Listagem 10.1 mostramos como serão inseridas as bibliografias no final do texto e também como fazer uma citação a um item da bibliografia no texto. Utilizamos o comando `\cite` para realizar a citação e como parâmetro devemos passar o rótulo do item da bibliografia que será citado no local. No exemplo dado, o rótulo foi `nono1878`. Este item de bibliografia deverá estar contido no arquivo `bibliografia.bib`, com o mesmo rótulo. Para ilustrar, a Listagem 10.2 apresenta um exemplo de item de bibliografia contido no arquivo `.bib`.

Listagem 10.2: Exemplo de um item de bibliografia

```
@book{nono1878,
    title = {The Principles of Nono},
    author = {John Nono},
    isbn = {123456789},
    year = {1878},
    publisher = {Magic Press}
}
```

10.3 O arquivo .bib

O arquivo `.bib` conterá os itens de bibliografia que poderão ser dos seguintes tipo: `article`, `book`, `inproceedings`, `incollection`, `inbook`, `booklet`, `phdthesis`, `mastersthesis`, `techreport`, `manual`, `misc`, `unpublished`, `online`. Cada um dos itens deverá ter um rótulo, pelo qual será referenciado no documento e terá diversos atributos para descrever o item, tais como: `author`, `title`, `journal`, `year`, `volume`, `number`, `pages`, `DOI`, `isbn`, `publisher`, `keywords`, `url`, dentre outros.

A Listagem 10.3 apresenta alguns exemplos de itens em um arquivo de referências bibliográficas.

Listagem 10.3: Exemplo de itens no arquivo `.bib`.

```
@Book{knuth1986,
  author = {Donald Knuth},
  title = {The TeXbook},
  publisher = {Addison-Wesley},
  year = {1986},
  address = {Reading, Mass},
  isbn = {9780201134483}
}

@book{lamport1994,
 Author = {Leslie Lamport},
 title = {LaTeX: A Document Preparation System},
 publisher = {Addison-Wesley Professional},
 year = {1994},
 month = {jul},
 isbn = {0201529831}
}

@article{einstein1905,
  author = "Albert Einstein",
  title = "{Zur Elektrodynamik bewegter K{\"o}rper}"
```

```
22    journal = "Annalen der Physik",
23    volume = "322",
24    number = "10",
25    pages = "891--921",
26    year = "1905"
27  }
```

Como pode ser visto nos exemplos apresentados na Listagem 10.3, existem vários campos para compôr um item de bibliografia. No campo `author`, você deverá preencher o nome do autor completo. Da mesma forma deverá proceder na campo `editor`. Os nomes podem ser escritos da forma simples, com ou sem vírgula, como apresentamos abaixo:

```
"Ludwig van Beethoven"
"van Beethoven, Ludwig"
```

Qualquer uma das formas funcionará adequadamente. O estilo de bibliografia escolhido fará a conversão do nome para o formato escolhido, quer utilize iniciais, nomes completos, caixa alta ou não. Isto é muito bom, pois caso seja necessário mudar o padrão, bastará trocar o estilo de bibliografia escolhido. Veja alguns exemplos na Seção 10.5. No caso dos campos `author` e `editor` possuírem mais de um nome, você deverá separá-los por `and`, por exemplo:

```
"Wolfgang Amadeus Mozart and Antonio Salieri"
```

Caso nomes, títulos ou qualquer informação na referência bibliográfica contenha acentos, estes deverão ser representados pelos comandos TeX para gerar os mesmos, colocando chaves ao redor do comando que produz o caractere acentuado. Por exemplo:

```
"Kurt G{\"{o}}del"          "Guimar{\~{a}}es Rosa"
```

A formatação do título também será definida pelo estilo escolhido, assim o BibTeX fará os ajustes de maiúsculas e minúsculas necessários. Caso alguma palavra deva ser escrita literalmente da forma como foi digitada, coloque-a entre chaves para manter a formação. Veja o exemplo:

```
"A Hist{\'{o}}ria da {NASA}"
```

10.4 Gerando a bibliografia

As Seções 10.2 e 10.3 mostram como criar um arquivo com as bibliografias e como citar os itens de bibliografia ao longo do texto. Se você gerar o

documento final usando o `latex` (para gerar o `dvi`) ou `pdflatex` (para gerar o `pdf`), verá que no local das citações aparecerão `?` (interrogação). O `latex` ou `pdflatex` irão criar, dentre outras coisas, uma tabela com as citações contidas no texto. Será necessário ainda utilizar a ferramenta `bibtex`, responsável pela automatização dos processos relativos às referências em uso nos arquivos para LaTeX. O `bibtex` fará a leitura da tabela das citações utilizadas no texto, realiza a busca na base de dados criada no arquivo `.bib` e por fim cria a bibliografia para o documento. Derá então rodar o comando `latex` ou `pdflatex` novamente para que a bibliografia seja incorporada ao documento. Lembre-se disso, o LaTeX está sempre gerando o texto com base no arquivo de bibliografia gerado na última vez que você chamou o `latex` ou `pdflatex` e o `bibtex`.

Uma outra ferramenta alternativa ao `bibtex` é o `biber`. O `biber` oferece mais funcionalidades e suporte a Unicode. Entretanto o `biber` não é tão difundido como o `bibtex`, talvez pela falta de suporte ao pacote `natbib`, um pacote de customização de citações.

10.5 Estilos

O LaTeX permite a escolha do estilo de bibliografia que será utilizada em seu documento. Para fazer isto insira o comando da Listagem 10.4 especificando no parâmetro o nome do estilo desejado.

Listagem 10.4: Escolhendo o estilo da bibliografia

```
\bibliographystyle{plain}
```

Apresentados aqui alguns estilos retirados da documentação do Overleaf. Você poderá encontrar mais demonstrações lá, no site do CTAN ou em outros lugares.

numeric

Items are cited: *The LaTeX Companion* book [2], the Einstein journal paper [1], and The LaTeX related items are [2, 3].

References

[1] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[2] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[3] Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

alphabetic

Items are cited: *The LaTeX Companion* book [GMS93], the Einstein journal paper [Ein05], and The LaTeX related items are [GMS93; Knu].

References

[Ein05] Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[GMS93] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[Knu] Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

authoryear

Items are cited: *The LaTeX Companion* book Goossens, Mittelbach, and Samarin 1993, the Einstein journal paper Einstein 1905, and The LaTeX related items are Goossens, Mittelbach, and Samarin 1993; Knuth n.d.

References

Einstein, Albert (1905). "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10, pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

Goossens, Michel, Frank Mittelbach, and Alexander Samarin (1993). *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley.

Knuth, Donald. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

authortitle

Items are cited: *The LaTeX Companion* book Goossens, Mittelbach, and Samarin, *The LaTeX Companion*, the Einstein journal paper Einstein, "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]", and The LaTeX related items are Goossens, Mittelbach, and Samarin, *The LaTeX Companion*; Knuth, *Knuth: Computers and Typesetting*.

References

Einstein, Albert. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

Goossens, Michel, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

Knuth, Donald. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

verbose

Items are cited: *The LaTeX Companion* book Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993, the Einstein journal paper Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004, and The LaTeX related items are Goossens, Mittelbach, and Samarin, *The LaTeX Companion*; Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

References

Einstein, Albert. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

Goossens, Michel, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

Knuth, Donald. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C~%7B%7Duno/abcde.html.

reading

Items are cited: *The LaTeX Companion* book Goossens, Mittelbach, and Samarin, *The LaTeX Companion*, the Einstein journal paper Einstein, "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]", and The LaTeX related items are Goossens, Mittelbach, and Samarin, *The LaTeX Companion*; Knuth, *Knuth: Computers and Typesetting*.

References

Einstein: Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies] einstein

Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

Goossens et al.: The LaTeX Companion latexcompanion

Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

Knuth: Knuth: Computers and Typesetting knuthwebsite

Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C-%7B%7Duno/abcde.html.

draft

Items are cited: *The LaTeX Companion* book **latexcompanion**, the Einstein journal paper **einstein**, and The LaTeX related items are **latexcompanion**; **knuthwebsite**.

References

einstein Albert Einstein. "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]". In: *Annalen der Physik* 322.10 (1905), pp. 891–921. DOI: http://dx.doi.org/10.1002/andp.19053221004.

knuthwebsite Donald Knuth. *Knuth: Computers and Typesetting*. URL: http://www-cs-faculty.stanford.edu/%5C-%7B%7Duno/abcde.html.

latexcompanion Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

apa

1 First section

Items that are cited: *The LaTeX Companion* book Goossens, Mittelbach, and Samarin, 1993, The Einstein's journal paper Einstein, 1905 and the Dirac's book Dirac, 1981 are physics related items. Next, a citation about *The LaTeX Companion* book Goossens et al., 1993.

References

Dirac, P. A. M. (yearmonthday). *The principles of quantum mechanics*. International series of monographs on physics. Clarendon Press.

Einstein, A. (yearmonthday). Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik*, *322*(10), 891–921. doi:http://dx.doi.org/10.1002/andp.19053221004

Goossens, M., Mittelbach, F., & Samarin, A. (yearmonthday). *The LaTeX companion*. Reading, Massachusetts: Addison-Wesley.

ieee

1 First section

Items that are cited: *The LaTeX Companion* book [1], The Einstein's journal paper [2] and the Dirac's book [3] are physics related items. Next, a citation about *The LaTeX Companion* book [1].

References

[1] M. Goossens, F. Mittelbach, and A. Samarin, *The LaTeX Companion*. Reading, Massachusetts: Addison-Wesley, 1993.

[2] A. Einstein, "Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]", *Annalen der Physik*, vol. 322, no. 10, pp. 891–921, 1905. DOI: http://dx.doi.org/10.1002/andp.19053221004.

[3] P. A. M. Dirac, *The Principles of Quantum Mechanics*, ser. International series of monographs on physics. Clarendon Press, 1981, ISBN: 9780198520115.

nature

1 First section

Items that are cited: *The LaTeX Companion* book [1], The Einstein's journal paper [2] and the Dirac's book [3] are physics related items. Next, a citation about *The LaTeX Companion* book [1].

References

1. Goossens, M., Mittelbach, F. & Samarin, A. *The LaTeX Companion* (Addison-Wesley, Reading, Massachusetts, 1993).
2. Einstein, A. Zur Elektrodynamik bewegter Körper. (German) [On the electrodynamics of moving bodies]. *Annalen der Physik* **322,** 891–921 (1905).
3. Dirac, P. A. M. *The Principles of Quantum Mechanics* ISBN: 9780198520115 (Clarendon Press, 1981).

10.6 Softwares de gerenciamento

Existem diversos softwares que facilitam o processo de inserir novas referências bibliográficas. Não há pretensão em citar todos os softwares existentes. Apenas serão citados alguns para exemplificar: 1. JabRef, 2. Mendeley, 3. BibDesk, 4. Papers, 5. KBitTex, 6. Zotero.

Para adicionar livros e artigos, existem sítios na internet que provêem ferramentas para facilitar. Por exemplo, o sítio `https://www.doi2bib.org/` fornece a entrada para BibTeX a partir do código DOI de um documento. Já o sítio `https://www.xarg.org/tools/isbn-to-bibtex/` fornece a entrada BibTeX, dado o código ISBN de um livro. O Google Books também fornece a entrada para BibTeX das obras que estão lá catalogadas. Procure na internet e encontrará ainda outras ferramentas.

Capítulo 11

Listagens

Sumário

11.1 Introdução 134
11.2 Listings 134
11.2.1 Inserindo Legenda 134
11.2.2 Criando Referência ao Código 135
11.2.3 Definindo a Linguagem 135
11.2.4 Configurações Gerais 135
11.2.5 Definindo Tamanho da Fonte 136
11.2.6 Comentários 136
11.2.7 Cor do Comandos 137
11.2.8 Numeração 138
11.2.9 Alterar o nome da Listagem 139
11.2.10 Quebra Automática de Linhas 140
11.2.11 Acentuação e Caracteres Especiais 141

11.1 Introdução

O LaTeXpossui suporte para inserção de códigos através dos pacotes `listings` e `minted`. Neste capítulo abordaremos os comandos para inserção de códigos através do uso de exemplos.

11.2 Listings

Para utilizar o pacote Listings é necessário carregar a biblioteca no início do documento como na Listagem 11.1.

Listagem 11.1: Inserindo Pacotes Listings

```
\usepackage{listings}
```

O código pode ser importado diretamente do seu disco rígido ou pode ser inserido diretamente no documento. Para inserir o código no documento utilize o exemplo da Listagem 11.2.

Listagem 11.2: Inserindo Código com Listings

```
\begin{lstlisting}
print("Oi Mundo!")
\end{lstlisting}
```

11.2.1 Inserindo Legenda

Para inserir legenda no código utilizamos o comando `caption` como na Listagem 11.3.

Listagem 11.3: Inserindo Código com Listings

```
\begin{lstlisting}[caption=Primeiro Exemplo]
print("Oi Mundo!")
\end{lstlisting}
```

11.2.2 Criando Referência ao Código

Para fazer referência ao código utilizamos o comando `label` como na Listagem 11.4. Neste caso para fazer referência ao código utilize o comando `ref`

Listagem 11.4: Criando Referências com Listings

```
A Listagem \ref{codigo1} apresenta o primeiro exemplo de um programa
em Python.
\begin{lstlisting}[caption=Primeiro Exemplo,label=codigo1]
print("Oi Mundo!")
\end{lstlisting}
```

11.2.3 Definindo a Linguagem

O usuário pode definir a linguagem de programação utilizada através do comando `language` como na Listagem 11.6. O Listings tem suporte a diversas Linguagens de Programação.

Listagem 11.5: Definindo a Linguagem de Programação

```
A Listagem \ref{codigo1} apresenta o primeiro exemplo de um programa
em Python.
\begin{lstlisting}[language=Python,caption=Primeiro Exemplo,
label=codigo1]
print("Oi Mundo!")
\end{lstlisting}
```

11.2.4 Configurações Gerais

Muitas vezes as configurações são as mesmas para todos os códigos no documento. O usuário pode definir estas configurações de maneira global no documento utilizando o comando `lstset`. A Listagem 11.7 apresenta como definir estas configurações. Neste exemplo definimos o background como amarelo e a linguagem como Python. Para a definição de cores funcionar é necessário inserir o pacote `xcolor`.

Listagem 11.6: Configurações Gerais

```
\usepackage{xcolor}
\usepackage{listings}
\lstset{
  backgroundcolor=\color{yellow},
  language=Python
}
```

11.2.5 Definindo Tamanho da Fonte

O usuário pode definir o tamanho da fonte através do comando `basicstyle` e todos os códigos terão o tamanho da fonte definido. Ele possui os seguintes estilos:

- footnotesize - Oi Mundo!!
- small - Oi Mundo!!
- itshape - *Oi Mundo!!*
- ttfamily - `Oi Mundo!!`
- scriptsize - Oi Mundo!!
- bfseries - **Oi Mundo!!**
- tiny - Oi Mundo!!

A Listagem 11.7 apresenta um exemplo utilizando código em itálico e em negrito.

Listagem 11.7: Configurações Gerais

```
\usepackage{xcolor}
\usepackage{listings}
\lstset{
  backgroundcolor=\color{yellow},
  language=Python,
  basicstyle = \itshape\bfseries
}
```

A Figura 11.1 apresenta o resultado da Listagem 11.7.

11.2.6 Comentários

Podemos alterar a cor dos comentários através do comando `commentstyle` como na Listagem 11.8. Vamos aproveitar e alterar a cor do `background`

Listing 1: Primeiro Exemplo

```
print("Oi Mundo!")
```

Figura 11.1: Resultado da Listagem 11.7

para deixar o exemplo mais completo. Vamos inserir uma cor cinza mais clara para facilitar a visualização utilizando o padrão do stackoverflow.

Listagem 11.8: Configurações Gerais

```
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily
}
```

A Figura 11.2 apresenta o resultado da Listagem 11.8.

Listing 1: Primeiro Exemplo

```
# Imprime Oi Mundo
print("Oi Mundo!")
```

Figura 11.2: Resultado da Listagem 11.8

11.2.7 Cor do Comandos

Podemos alterar a cor reservada para os comandos da linguagem de programação. O comando `keywordstyle` pode ser utilizado para alterar as cores dos comandos. A Listagem 11.9 apresenta um exemplo do uso deste comando.

Listagem 11.9: Configurações Gerais

```
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
```

```
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily,
  keywordstyle=\color{blue}
}
```

A Figura 11.3 apresenta o resultado da Listagem 11.9.

Listing 1: Primeiro Exemplo

```
# Imprime Oi Mundo
print("Oi Mundo!")
```

Figura 11.3: Resultado da Listagem 11.9

11.2.8 Numeração

O usuário pode escolher várias opções em relação a numeração:

- numbers - posição da numeração (none, left, right)
- numbersep = distância dos números do código
- numberstyle - estilos dos números (fonte)
- stepnumber - incremento da numeração

Listagem 11.10: Configurações Gerais

```
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily,
  keywordstyle=\color{blue},
  numbers=left,
  numbersep=3pt,
```

```
  numberstyle=\tiny\color{red},
  stepnumber=1
}
```

A Figura 11.4 apresenta o resultado da Listagem 11.10.

Listing 1: Primeiro Exemplo

```
# Imprime Oi Mundo
print("Oi Mundo!")
```

Figura 11.4: Resultado da Listagem 11.10

11.2.9 Alterar o nome da Listagem

O usuário pode alterar o nome da Listagens que até então em nossos exemplos era Listing para outro nome. Outra possibilita é criar uma lista com todas as listagens. A Listagem 11.12 apresenta o exemplo realizando estas duas alterações.

Listagem 11.11: Configurações Gerais

```
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
\renewcommand{\lstlistingname}{Código}
\renewcommand{\lstlistlistingname}{Lista de  \lstlistingname s}
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily,
  keywordstyle=\color{blue},
  numbers=left,
  numbersep=3pt,
  numberstyle=\tiny\color{red},
  stepnumber=1
}
```

A Figura 11.5 apresenta o resultado da Listagem 11.12.

Lista de Códigos

1 Primeiro Exemplo 1

1 Listings

Código 1: Primeiro Exemplo

```
# Imprime Oi Mundo
print("Oi Mundo!")
```

Figura 11.5: Resultado da Listagem 11.12

11.2.10 Quebra Automática de Linhas

O usuário pode optar pela quebra automática de linhas quando o código inserido ficar maior que a área de texto disponível. O comando `breaklines` tem a opção de argumentos true ou false. A Listagem 11.13 apresenta o uso deste comando.

Listagem 11.12: Quebra Automática de Linhas

```latex
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
\renewcommand{\lstlistingname}{Código}
\renewcommand{\lstlistlistingname}{Lista de  \lstlistingname s}
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily,
  keywordstyle=\color{blue},
  numbers=left,
  numbersep=3pt,
  numberstyle=\tiny\color{red},
  stepnumber=1,
  breaklines=true
}
```

11.2.11 Acentuação e Caracteres Especiais

Quando utilizamos caracteres especiais ou acentuação não conseguiremos compilar o código. Uma opção é utilizar o código da Listagem 11.13 para solucionar o problema.

Listagem 11.13: Acentuação e Caracteres Especiais

```
\usepackage{listings}
\usepackage{xcolor}
\usepackage{color}
\renewcommand{\lstlistingname}{Código}
\renewcommand{\lstlistlistingname}{Lista de  \lstlistingname s}
\definecolor{light-gray}{gray}{0.92}
\lstset{
  backgroundcolor=\color{light-gray},
  language=Python,
  basicstyle = \small,
  commentstyle=\color{orange}\ttfamily,
  keywordstyle=\color{blue},
  numbers=left,
  numbersep=3pt,
  numberstyle=\tiny\color{red},
  stepnumber=1,
   literate=%
            {à}{{\`a}}1
            {á}{{\'a}}1
            {ã}{{\~a}}1
            {â}{{\^a}}1
            {é}{{\'e}}1
            {ê}{{\^e}}1
            {í}{{\'i}}1
            {ó}{{\'o}}1
            {õ}{{\~o}}1
            {ú}{{\'u}}1
            {ü}{{\"u}}1
            {ç}{{\c{c}}}1
         {í}{{\'i}}1
         {é}{{\'e}}1
         {ý}{{\'y}}1
         {ú}{{\'u}}1
         {ó}{{\'o}}1
         {ě}{{\v{e}}}1
         {š}{{\v{s}}}1
         {č}{{\v{c}}}1
```

```
        {ř}{{\v{r}}}1
        {ž}{{\v{z}}}1
        {ď}{{\v{d}}}1
        {ť}{{\v{t}}}1
        {ň}{{\v{n}}}1
        {ů}{{\r{u}}}1
        {Á}{{\'A}}1
        {Í}{{\'I}}1
        {É}{{\'E}}1
        {Ý}{{\'Y}}1
        {Ú}{{\'U}}1
        {Ó}{{\'O}}1
        {Ě}{{\v{E}}}1
        {Š}{{\v{S}}}1
        {Č}{{\v{C}}}1
        {Ř}{{\v{R}}}1
        {Ž}{{\v{Z}}}1
        {Ď}{{\v{D}}}1
        {Ť}{{\v{T}}}1
        {Ň}{{\v{N}}}1
        {Ů}{{\r{U}}}1
        {Ö}{{\"O}}1
    {Ä}{{\"A}}1
    {Ü}{{\"U}}1
    {ß}{{\ss}}1
    {ü}{{\"u}}1
    {ä}{{\"a}}1
    {ï}{{\"i}}1
    {ö}{{\"o}}1
    {~}{{\textasciitilde}}1
}
```

A Figura 11.6 apresenta o resultado da Listagem 11.13.

Código 1: Primeiro Exemplo

```
# Imprime Oi Mundo
print("Oi Mundo!")
print("Acentuação")
```

Figura 11.6: Resultado da Listagem 11.13

Capítulo 12

Erros

Sumário

12.1 Erros **144**
12.2 Erros comuns **144**
12.3 Solucionando erros **146**

12.1 Erros

Infelizmente não somos a prova de erros. Cometermos muitos deslizes quando estamos escrevendo nosso texto com LaTeX. É importante então saber lidar com eles e buscar evitá-los. Ao compilar o documento o LaTeX poderá se deparar com erros e fornecerá uma mensagem alertando sobre o mesmo. Algumas vezes essas mensagens podem não ajudar muito na localização do erro. O que pode ocorrer nesses casos é um erro em um determinado local, mas que o LaTeX não imagina que aquilo pode ser um erro, então ele continua processando o documento até chegar num determinado ponto onde as coisas passam a não fazer mais sentido e desmorona. Imagine que você tenha esquecido de fechar uma chaves. A ausência deste simples caractere pode acabar gerando um erro em outro local, afinal, o LaTeX não tem como adivinhar que você esqueceu-se de fechar uma chaves e continua processa tudo que aparece em sequência como parte de uma mesmo contexto. Eventualmente aquele pequeno erro vai acarretar em um erro catastrófico em algum outro lugar.

Uma boa prática para evitar que os erros virem monstros é manter duas boas práticas: 1. modularizar o documento; 2. compilar o documento pouco a pouco, não aguarde escrever páginas e páginas para gerar o seu documento, assim ficará mais fácil localizar o erro, você saberá que ele foi inserido na parte do texto que foi editada por último ou poderá comentar facilmente blocos do texto para mais rapidamente localizá-los.

12.2 Erros comuns

Nesta seção iremos listar alguns erros comuns, geralmente causado por algum descuido.

alinhamento mensagem de erro: `Misplaced alignment tab character &`. Este erro ocorre quando utilizamos o caractere de alinhamento `&` de forma incorreta, geralmente em algum nome próprio que contém `&`. Devemos lembrar de converter `&` na sequência `\&`. Outro possível erro de alinhamento é a falta de um `&` em uma tabela. Neste caso será dada a mensagem de erro: `Extra alignment tab has been changed to \cr`. Este tipo de erro é comum quando estamos lidando com uma tabela grande. Busque sempre que possível deixar a tabela com as células alinhas no código, assim ficará mais fácil ver se está faltando ou se existe um `&` extra.

modo matemático mensagem de error: `Missing $ inserted`. Devemos

observar que as equações, quando usamos o `$` para marcá-las, devem estar na forma `$ . . . $` ou `$$ . . . $$`. Caso esqueça algum dos `$`, irá observar este erro. Devemos lembrar também de fechar corretamente as chaves usadas para agrupar itens. Caso esqueça de fechar uma delas, irá ver a mensagem de erro `Missing } inserted`. Outro erro comum é esquecer de fechar parênteses, colchetes ou chaves quando usando o modo `\left`. Nestes casos será gerada a mensagem de erro `Missing \right. inserted`.

referências uma referência vazia ou errada gera uma mensagem de aviso `Warning: Reference ‘’` e no texto aparecerá `??` no local. Neste caso não é um erro propriamente dito, mas apenas um aviso. Entretanto se estiver utilizando o pacote `cleveref`, uma referência varia gera um erro com a mensagem `Argument of \@cref@stack@top has an extra }`, impedindo o documento de ser gerado. Outro erro comum é copiar e colar algum trecho de código e esquecer de trocar o rótulo (*label*). Isto acarretará no aviso `Warning: Label ‘xxx’ multiply defined.`.

comandos não identificados é comum o erro de digitar um comando LaTeX, errando assim a mensagem de erro `Undefined control sequence`. Este erro também poderá ocorrer se você estiver utilizando um comando que foi definido em um pacote e esqueceu de incluir o pacote, ou ainda se não definiu o comando como esperado no preâmbulo do documento.

arquivo não encontrado pode ocorrer de inserimos um arquivo e este não existir, quer o nome tenha sido digitado errado, o caminho esteja errado ou realmente não exita o arquivo. Quando isso correr, será gerada a mensagem de erro `Error: File ‘xxxxxxx’ not found`. Isto pode ocorrer quando inserimos uma imagem através do `includegraphics` ou um outro arquivo `.tex` através do `input`.

comprimento quando utilizamos algum comando que recebe como parâmetro uma medida de comprimento, como por exemplo os comandos `hspace` e `vspace`, se o argumento estiver mal formatado ou conter um valor ilegal, será gerada uma das mensagens: `Missing number, treated as zero` ou `Illegal unit of measure (pt inserted)`.

ambiente é comum também inserirmos algum código contendo algum ambiente, sendo que este depende de algum pacote que não foi adicionado no preambulo. Neste caso, será gerada a mensagem de erro `LaTeX Error: Environment xxxxxx undefined`.

inclusão fora do preâmbulo caso você tenha tentado adicionar um pacote fora do preâmbulo, será gerada a mensagem de erro `Error: Can be used only in preamble`.

sem item uma lista sem utilizar o comando `item` irá causar o erro `Error: Something's wrong--perhaps a missing \item`.

12.3 Solucionando erros

Como dito anteriormente, na Seção 12.1, a melhor forma de evitar e solucionar erros que venham a ocorrer é constantemente compilar o documento para verificar se não houve algum erro desde a última vez que você gerou o documento. Fique sempre atento à sintaxe, à digitação dos comandos e ao uso de caracteres que possuem significado especial. Elaborar o documento utilizando um editor que tenha recurso de marcar com cores os comandos e tenha recurso de autocompletar os comandos que são digitados irá diminuir as chances de você cometer algum erro.

Sempre que necessário, volte à documentação para certificar se está utilizando os comandos da forma correta. Caso não consiga resolver sozinho o erro, poderá buscar na internet pela mensagem de erro e provavelmente encontrará alguém que passou pelo mesmo problema e já achou a solução. Em último caso, não achando a solução, poderá recorrer às comunidades online. A comunidade no `StackExchange` é bem ativa. Lembre-se sempre de seguir as regras da comunidade.

Índice

Índice Remissivo
    index, 122
    makeidx, 122
    makeindex, 123
    printindex, 122
TEX, 26

Bibliografia
    bibliography, 47, 126
    bibliographystyle, 47, 126
book, 46

chapter, 46
Comandos
    ldots, 100

Erros
    LaTeX Error: Too deeply nested, 92
espaço, 32

Figuras
    figure, 66
    height, 72
    scale, 73

Git, 29

Linux, 26
Listas
    description, 99
    enumerate, 93, 94
    item, 90
    itemize, 90–92
    label, 91

Mac OS X, 26, 28
Miktex, 27

Overleaf, 29

Pacotes
    enumitem, 91, 94
    graficx, 66
    multicol, 101
    subfigure, 68, 75
part, 46

report, 46

Shell
    Shell, 26
Sistema Operacional, 26
Sistemas Operacionais
    Windows, 27
Softwares, 26
Sumário
    tableofcontents, 47

Título
    maketitle, 47
Texmaker, 29
Tipos de Documentos
    book, 47

Vim, 29

Windows, 26

Bibliografia

[1] Beebom. 10 Best LaTeX Editors You Should Use.

[2] Kile. Kile.

[3] Donald Knuth. *The TeXbook*. Addison-Wesley, Reading, Mass, 1986.

[4] Donald E. Knuth. *Typesetting Concrete Mathematics*, volume 10. TUGboat, 1989.

[5] Donald E. Knuth and Hermann Zapf. *AMS Euler: A New Typeface for Mathematics*, volume 20. Scholarly Publishing, 1989.

[6] Richard Koch. TexShop.

[7] MacTeX Technical Working Group. Mac Tex.

[8] Overleaf. Overleaf.

[9] Henry Walter Pendry. *The Baudôt Printing Telegraph System*. Sir Isaac Pitman and Sons, Ltd., London, 1919.

[10] Marko Pinteric. Using latex on windows. `http://www.pinteric.com/miktex.html`, 2004.

[11] Texmaker. TeXmaker.

[12] TeXstudio. TeXstudio.

[13] TexWorks. TexWorks.

[14] Wikipedia. Comparison of reference management software — Wikipedia, the free encyclopedia. `http://en.wikipedia.org/w/index.php?title=Comparison%20of%20reference%20management%20software&oldid=917926353`, 2019. [Online; accessed 27-September-2019].

View publication stats