

**Inescapable bias**  
*The role of recommender systems in social  
media radicalization*

Caio Lente

REPORT PRESENTED TO THE  
INSTITUTE OF MATHEMATICS AND STATISTICS  
OF THE UNIVERSITY OF SÃO PAULO  
FOR THE MASTER OF SCIENCE  
QUALIFYING EXAMINATION

Program: Computer Science

Advisor: Prof. Dr. Roberto Hirata Jr.

São Paulo  
August 10th, 2017



## **Inescapable bias**

### ***The role of recommender systems in social media radicalization***

Caio Lente

This is the original version of the qualifying  
text prepared by the candidate Caio Lente,  
as submitted to the Examining Committee.

I hereby authorize the reproduction and distribution in full or in part of this work, in any conventional or electronic medium, for study or research, as long as properly cited.

*To my four parents and one and a half siblings.*



## Resumo

Caio Lente. **Viés inescapável: O papel de sistemas de recomendação na radicalização das mídias sociais**. Exame de Qualificação (Mestrado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2021.

[illegible]

**Palavras-chave:** sistemas de recomendação. viés algorítmico. aprendizagem de máquina.





# Abstract

Caio Lente. **Inescapable bias: *The role of recommender systems in social media radicalization***. Qualifying Exam (Masters). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2021.

[illegible]

**Keywords:** recommender system, algorithmic bias, machine learning.



## List of Figures

4.1	Exemplo de subfiguras. . . . .	26
4.2	Exemplo de cronograma. . . . .	27
4.3	Exemplos de gráficos gerados externamente . . . . .	27

## List of Tables

4.1	Exemplos de tabelas (códigos, abreviaturas e nomes dos aminoácidos). . .	28
4.2	Exemplo de tabela com valores numéricos. . . . .	28
4.3	Exemplo de tabela similar a uma ficha. . . . .	30

## List of Programs

4.1	Exemplo de laço em Java. . . . .	27
A.1	Máximo divisor comum (arquivo importado). . . . .	31
A.2	Máximo divisor comum. . . . .	31



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Social Networks . . . . .	2
1.2	Recommender Systems . . . . .	3
1.3	Radicalization . . . . .	4
1.4	Hypothesis . . . . .	4
1.5	Considerações de Estilo . . . . .	5
1.6	Ferramentas Bibliográficas . . . . .	6
1.7	O Que o IME Espera . . . . .	7
<b>2</b>	<b>Usando o <math>\LaTeX</math> e este modelo</b>	<b>9</b>
2.1	Instalação do $\LaTeX$ . . . . .	10
2.2	Documentação sobre $\LaTeX$ . . . . .	11
<b>3</b>	<b>Do zero ao mínimo com <math>\LaTeX</math></b>	<b>13</b>
3.1	Visão Geral . . . . .	14
3.2	Estrutura de um Documento $\LaTeX$ . . . . .	15
3.3	Executando $\LaTeX$ e Comandos Auxiliares . . . . .	16
3.4	Mais sobre Estrutura . . . . .	16
3.5	Referências Cruzadas e <i>Floats</i> . . . . .	17
3.6	Fórmulas Matemáticas . . . . .	18
3.7	Referências Bibliográficas e Bibliografia . . . . .	18
3.8	Imagens, Ilustrações, Diagramas e Gráficos . . . . .	20
3.9	Formatação Manual . . . . .	20
3.10	Detalhes da Linguagem . . . . .	21
3.11	Versões do $\LaTeX$ . . . . .	22
3.12	Limitações do $\LaTeX$ e Algumas Dicas . . . . .	23
<b>4</b>	<b>Alguns Exemplos de Comandos <math>\LaTeX</math></b>	<b>25</b>
4.1	Bibliografia e Referências . . . . .	25

4.2	Modo Matemático . . . . .	25
4.3	<i>Floats</i> (Tabelas e Figuras) . . . . .	26

## **Appendices**

<b>A</b>	<b>Código-Fonte e Pseudocódigo</b>	<b>31</b>
----------	------------------------------------	-----------

## **Annexes**

<b>A</b>	<b>Perguntas Frequentes sobre o Modelo</b>	<b>33</b>
----------	--	-----------

	<b>References</b>	<b>35</b>
--	-------------------	-----------

# Chapter 1

## Introduction

Social networks have all but taken over contemporary daily life. From the eponymous socializing, to reading news, to expressing ourselves, social media has crept into every corner of society. Most of its side-effects, it could be argued, are positive (shortening distances, political accountability, social organizing), but they are not perfect institutions.

Social media companies already face significant backlash for their questionable business model and ethics. Cambridge Analytica's election meddling, Facebook's subliminal experiments, YouTube's problem with disturbing content marketed at kids, and Twitter's bot infestation are just a few recent scandals that have put the societal role of social media into question.

One particular controversy that has taken over public discourse around social networks is the role that their algorithms might have in radicalizing users, specially younger ones. The aforementioned experiments conducted by Facebook to influence people's emotions and the proliferation of more than questionable videos aimed at children on YouTube are instances that seem to corroborate the notion that there is something fundamentally wrong with these companies' algorithms.

News organizations, in general, have been skeptical of social networks. Journalists and specialists alike argue that social media's algorithms (specially recommender algorithms) are tuned to peddle conspiracy theories, extremist views, and false information. This would be the source cause for a plethora of what they consider contemporary evils: religious extremism, anti-democratic leaders, widespread depression among teenagers, anti-science movements, etc.

This narrative, of course, has been questioned for a variety of reasons. Some say that it is self serving: traditional news organizations are being displaced by social media and it would be convenient for them to mine the public's trust in them. Others claim that these recommender algorithms are not to blame for political polarization and that social networks even have a tendency to favor more left-wing viewpoints.

The debate around the role of recommender systems in social media radicalization is still, unfortunately, too recent and based in anecdotes. Since its impacts are all but

universal, more quality research is vital to inform both the public and opinion makers about if and how much recommendation algorithms influence social media users.

This dissertation aims to further such research. The rest of Chapter 1 is dedicated to core concepts covered in the rest of the work, ending in subsection 1.4, which tackles the main hypothesis of this dissertation. Chapter 2 contains the literature review, Chapter 3 explains the experiments already conducted and their results, and Chapter 4 is about next steps.

## 1.1 Social Networks

Social networking services, also referred to as social networks and social media, are notoriously difficult to define. Some definitions might be too narrow (excluding instant messaging services), while some might be too broad (including technologies such as telephone networks). Most definitions include some common features:

- Internet-based
- Focus on user-generated content
- Users have profiles
- Users can connect

While social-networking-like applications already existed in Usenet, Geocities, launched in 1994, is usually regarded as the first major social network. Friendster and Myspace followed in 2003, with Orkut and Facebook slightly lagging behind in 2004. Each hit their peak at different moments and different countries, but Facebook overtook all of them in 2009 when it became the most popular social networking service in the world, still maintaining the title over 11 years later at the moment of writing.

Even though all aforementioned social networks are multimedia, that is, users can post text, photos and videos, some of the most popular services focus on a specific type of media. For instance, YouTube (2009) centers around videos, WhatsApp (2009) and WeChat (2011) were originally designed for text-based communication, and Instagram's (2010) main focus is photos.

Some social media services, very much in agreement with McLuhan's teachings, have what could be considered a "style". Instagram's content, for example, tends toward more personal (i.e. egoic) photos and videos. As of November 28th, 2020, six of the top 20 most-liked posts on Instagram are from american socialite Kylie Jenner, consisting of four photos of her daughter and two of her ex-boyfriend. Even though there are many different niches inside Instagram, personal posts seem to have an edge over other kinds of content.

Twitter, unlike most other social networks, allows for asymmetrical connections, meaning users can follow profiles without being followed back. This enables the emergence of Twitter communities (e.g. Fintwit, Black Twitter) that can be largely self referential and/or organized around certain subjects. Facebook users, on the other hand, can belong to groups, user-moderated profiles that might revolve around any particular topics of interest; there are groups that organize pet owners and groups that organize neonazis.



Parallel to all other features and idiosyncrasies, there lay the recommendation algorithms. While a few social networking services (e.g. WhatsApp) do not recommend any content or profiles to the user, most do and, according to recent studies, these recommendations have become the main drivers of interactions.

## 1.2 Recommender Systems

Recommender systems (sometimes called recommendation systems or recomender algorithms) first appeared in 1992 under the name “collaborative filtering”, even though that term nowadays refers to a subclass of recommender systems. The aim of such an algorithm is providing users with personalized product or service recommendations, an essential task when considering the ever increasing number of possible videos to watch, music to listen, products to buy.

The input of a recommender system is usually information about the preferences (ratings, likes/dislikes, watch time, etc.) of consumers for a set of items. Preference information can be gathered from explicit behaviors (e.g. rating a product in a scale ranging from 0 to 5 stars) or from implicit behaviors (e.g. how much time the user lingers on a product’s page). These data can be combined with information about the user (age, political leaning, etc.) in order to create the best possible representation of the user’s preferences.

The output of these systems can come in the form of a prediction or a list of recommended items. In the first case, the goal of the algorithm is approximating the rating a user would attribute to a yet unrated item, while the second type of output involves gathering the items that most likely would interest the user. Simple recommender systems that suggest items similar to the one being queried do not necessarily involve rating predictions, but it is common to have the list of recommended items based on the ratings the algorithms estimated the user would give to those items.

Most recommender systems follow into one of four categories according to the filtering algorithm they use, that is, the strategy for generating predictions or selecting the top-N items: content-based filtering, demographic filtering, collaborative filtering, and hybrid filtering.

Content-based filtering leverages characteristics of the content in order to generate the recommendations. One such algorithm might use the genres of watched movies in order to recommend new ones, while another might analyse the sound signature of a song to recommend similar ones, but, either way, all content-based systems establish a similarity between items as a basis for recommendations. Analogously, demographic filtering uses demographic data to establish a similarity between users and recommend items positively rated by similar people.

Collaborative filtering algorithms also recommend items that similar users liked, but, in this case, the similarity between users is based on past ratings and not demographic information. Hybrid filtering usually mix collaborative methods with either content-based or demographic filtering.

As with other knowledge-based systems, recommendation algorithms have quickly incorporated neural networks and other machine learning techniques over the past few

years. Even though the implementation of YouTube's recommendation algorithm is a trade secret, it is known to gather enormous amounts of data about the user's interaction with the website and to require Google's own TPUs in order to be trained. It also involves two distinct steps: candidate generation (when the billions of videos available on the platform are quickly narrowed down to a few hundreds that might be relevant) and ranking (when the algorithm actually attempts to predict the score a user would implicitly give to the candidate videos).

Another relevant aspect of recommender systems that is well-exemplified by YouTube is the use of balancing factors such as novelty, dispersity, and stability. In the case of Google's video giant, there is a baked-in bias for recency, strongly favoring newer videos in detriment of older content.

### 1.3 Radicalization

Opinion polarization is far from a recent phenomenon, and social media is only the most recent communication medium where it can be detected and studied. An important question is whether it facilitates or attenuates polarization: anecdotal evidence might suggest that social network structures incentivize users to gather into antagonistic communities, but this could be a result of people simply being more likely to express their preferences online, not of some intrinsic property of social media.

One possible byproduct of polarization is radicalization. Despite not being entirely different phenomena, these concepts deserve distinct levels of attention. While polarization can be considered a natural part of democratic discourse, radicalization only happens when certain conditions are met. UNESCO defines radicalization as:

- The individual person's search for fundamental meaning, origin and return to a root ideology;
- The individual as part of a group's adoption of a violent form of expansion of root ideologies and related oppositionist objectives;
- The polarization of the social space and the collective construction of a threatened ideal 'us' against 'them,' where the others are dehumanized by a process of scapegoating.

### 1.4 Hypothesis

Escrever bem é uma arte que exige muita técnica e dedicação e, consequentemente, há vários bons livros sobre como escrever uma boa dissertação ou tese. Um dos trabalhos pioneiros e mais conhecidos nesse sentido é o livro de Umberto Eco (2009) intitulado *Como se faz uma tese*; é uma leitura bem interessante mas, como foi escrito em 1977 e é voltado para trabalhos de graduação na Itália, não se aplica tanto a nós.

Sobre a escrita acadêmica em geral, John Carlis disponibilizou um texto curto e interessante (CARLIS, 2009) em que advoga a preparação de um único rascunho da tese antes da versão final. Mais importante que isso, no entanto, são os vários *insights* dele sobre a escrita

acadêmica. Dois outros bons livros sobre o tema são *The Craft of Research* (BOOTH *et al.*, 2008) e *The Dissertation Journey* (ROBERTS, 2010). Além disso, a USP tem uma compilação de normas relativas à produção de documentos acadêmicos (SIBrUSP, 2009) que pode ser utilizada como referência.

Para a escrita de textos especificamente sobre Ciência da Computação, o livro de Justin Zobel, *Writing for Computer Science* (ZOBEL, 2004) é uma leitura obrigatória. O livro *Metodologia de Pesquisa para Ciência da Computação* de Raul Sidnei WAZLAWICK (2009) também merece uma boa lida. Já para a área de Matemática, dois livros recomendados são o de Nicholas Higham, *Handbook of Writing for Mathematical Sciences* (HIGHAM, 1998) e o do criador do  $\text{\TeX}$ , Donald Knuth, juntamente com Tracy Larrabee e Paul Roberts, *Mathematical Writing* (KNUTH *et al.*, 1996).

Apresentar os resultados de forma simples, clara e completa é uma tarefa que requer inspiração. Nesse sentido, o livro de Edward TUFTE (2001), *The Visual Display of Quantitative Information*, serve de ajuda na criação de figuras que permitam entender e interpretar dados/resultados de forma eficiente.

Além desse material, também vale muito a pena a leitura do trabalho de Uri ALON (2009), no qual apresenta-se uma reflexão sobre a utilização da Lei de Pareto para tentar definir/escolher problemas para as diferentes fases da vida acadêmica. A direção dos novos passos para a continuidade da vida acadêmica deveria ser discutida com seu orientador.

## 1.5 Considerações de Estilo

Normalmente, as citações não devem fazer parte da estrutura sintática da frase<sup>1</sup>. No entanto, usando referências em algum estilo autor-data (como o estilo plainnat do  $\text{\LaTeX}$ ), é comum que o nome do autor faça parte da frase. Nesses casos, pode valer a pena mudar o formato da citação para não repetir o nome do autor; no  $\text{\LaTeX}$ , isso pode ser feito usando os comandos `\citet`, `\citep`, `\citeyear` etc. documentados no pacote natbib (DALY, 2010) (esses comandos são compatíveis com biblatex usando a opção `natbib=true`, ativada por padrão neste modelo). Em geral, portanto, as citações devem seguir estes exemplos:

Modos de citação:

indesejável: [AF83] introduziu o algoritmo ótimo.

indesejável: (Andrew e Foster, 1983) introduziram o algoritmo ótimo.

certo: Andrew e Foster introduziram o algoritmo ótimo [AF83].

certo: Andrew e Foster introduziram o algoritmo ótimo (Andrew e Foster, 1983).

certo (`\citet` ou `\citeyear`): Andrew e Foster (1983) introduziram o algoritmo ótimo.

O uso desnecessário de termos em língua estrangeira deve ser evitado. No entanto, quando isso for necessário, os termos devem aparecer *em itálico*.

Uma prática recomendável na escrita de textos é descrever as legendas das figuras e tabelas em forma auto-contida: as legendas devem ser razoavelmente completas, de modo que o leitor possa entender a figura sem ler o texto onde a figura ou tabela é citada.

---

<sup>1</sup>E não se deve abusar das notas de rodapé.

Sugerimos que você faça referências bibliográficas de forma similar aos estilos “alpha” (referências alfanuméricas) ou “plainnat” (referências por autor-data) de  $\text{\LaTeX}$ . Se estiver usando natbib+bibtex, use os arquivos .bst “alpha-ime.bst” ou “plainnat-ime.bst”, que são versões desses dois formatos traduzidas para o português. Se estiver usando biblatex (recomendado), escolha o estilo “alphabetic” (que é um dos estilos padrão do biblatex) ou “plainnat-ime”. O arquivo de exemplo inclui todas essas opções; basta des-comentar as linhas correspondentes e, se necessário, modificar o arquivo Makefile para chamar o bibtex ao invés do biber (este último é usado em conjunto com o biblatex).

## 1.6 Ferramentas Bibliográficas

Embora seja possível pesquisar por material acadêmico na Internet usando sistemas de busca “comuns”, existem ferramentas dedicadas, como o Google Scholar ([scholar.google.com](https://scholar.google.com)). Você também pode querer usar o Web of Science ([webofscience.com](https://webofscience.com)) e o Scopus ([scopus.com](https://scopus.com)), que oferecem recursos sofisticados e limitam a busca a periódicos com boa reputação acadêmica. Essas duas plataformas não são gratuitas, mas os alunos da USP têm acesso a elas através da instituição. Ambas são capazes de exportar os dados para o formato .bib, usado pelo  $\text{\LaTeX}$ . Algumas editoras, como a ACM e a IEEE, também têm sistemas de busca bibliográfica.

Apenas uma parte dos artigos acadêmicos de interesse está disponível livremente na Internet; os demais são restritos a assinantes. A CAPES assina um grande volume de publicações e disponibiliza o acesso a elas para diversas universidades brasileiras, entre elas a USP, através do seu portal de periódicos ([periodicos.capes.gov.br](https://periodicos.capes.gov.br)). Existe uma extensão para os navegadores Chrome e Firefox ([www.infis.ufu.br/capes-periodicos](http://www.infis.ufu.br/capes-periodicos)) que facilita o uso cotidiano do portal.

Para manter um banco de dados organizado sobre artigos e outras fontes bibliográficas relevantes para sua pesquisa, é altamente recomendável que você use uma ferramenta como Zotero ([zotero.org](https://zotero.org)) ou Mendeley ([mendeley.com](https://mendeley.com)). Ambas podem exportar seus dados no formato .bib, compatível com  $\text{\LaTeX}$ . Também existem três plataformas gratuitas que permitem a busca de referências acadêmicas já no formato .bib:

- *CiteULike* (patrocinados por Springer): [www.citeulike.org](http://www.citeulike.org)
- Coleção de bibliografia em Ciência da Computação: [liinwww.ira.uka.de/bibliography](http://liinwww.ira.uka.de/bibliography)
- Google acadêmico (habilitar bibtex nas preferências): [scholar.google.com](https://scholar.google.com)

Lamentavelmente, ainda não existe um mecanismo de verificação ou validação das informações nessas plataformas. Portanto, é fortemente sugerido validar todas as informações de tal forma que as entradas bib estejam corretas.

De qualquer modo, tome muito cuidado na padronização das referências bibliográficas: ou considere TODOS os nomes dos autores por extenso, ou TODOS os nomes dos autores abreviados. Evite misturas inapropriadas.

## 1.7 O Que o IME Espera

Ao terminar sua tese/dissertação, você deve entregar uma cópia dela para a CPG. Após a defesa, você tem 30 dias para revisar o texto e incorporar as sugestões da banca. Assim, há duas versões oficiais do documento: a versão original e a versão corrigida, o que deve ser indicado na folha de rosto.

Fica a critério do aluno definir aspectos como o tamanho de fonte, margens, espaçamento, estilo de referências, cabeçalho, etc. considerando sempre o bom senso. A CPG, em reunião realizada em junho de 2007, aprovou que as teses/dissertações deverão seguir o formato padrão por ela definido<sup>2</sup>. Esse padrão refere-se aos itens que devem estar presentes nas teses/dissertações (e.g. capa, formato de rosto, sumário, etc.), e não à formatação do documento. Ele define itens obrigatórios e opcionais, conforme segue:

- CAPA (obrigatória)
  - O IME usa uma capa padrão de cartolina para todas as teses/dissertações. Essa capa tem uma janela recortada por onde se vê o título e o autor do trabalho e, portanto, a capa impressa do trabalho deve incluir o título e o autor na posição correspondente da página. Ela fica centralizada na página, tem 100mm de largura, 60mm de altura e começa 47mm abaixo do topo da página.
  - O título da tese/dissertação deverá começar com letra maiúscula e o resto deverá ser em minúsculas, salvo nomes próprios.
  - O nome do aluno(a) deverá ser completo e sem abreviaturas.
  - É preciso explicitar se é uma tese ou dissertação (para obtenção do título de doutor, tese; para obtenção do título de mestre, dissertação).
  - O nome do programa deve constar da capa (Matemática, Matemática Aplicada, Estatística ou Ciência da Computação).
  - Também devem constar o nome completo do orientador e do co-orientador, se houver.
  - Se o aluno recebeu bolsa, deve-se indicar a(s) agência(s).
  - É preciso informar o mês e ano do depósito ou da entrega da versão corrigida.
- FOLHA DE ROSTO (obrigatória, tanto para a versão depositada quanto para a versão corrigida)
  - o título da tese/dissertação deverá seguir o padrão da capa
  - deve informar se se trata da versão original ou da versão corrigida; no segundo caso, deve também incluir os nomes dos membros da banca.
- AGRADECIMENTOS (opcional)
- RESUMO, em português (obrigatório)
- ABSTRACT, em inglês (obrigatório)

---

<sup>2</sup>[www.ime.usp.br/dcc/pos/normas/tesesedissertacoes](http://www.ime.usp.br/dcc/pos/normas/tesesedissertacoes)

- SUMÁRIO (obrigatório)
- LISTAS (opcionais)
  - Lista de Abreviaturas
  - Lista de Símbolos
  - Lista de Figuras
  - Lista de Tabelas
- REFERÊNCIAS BIBLIOGRÁFICAS (obrigatório)
- ÍNDICE REMISSIVO (opcional<sup>3</sup>)

---

<sup>3</sup>O índice remissivo pode ser muito útil para a banca; assim, embora seja um item opcional, recomendamos que você o crie.

## Chapter 2

# Usando o L<sup>A</sup>T<sub>E</sub>X e este modelo

Não é necessário que o texto seja redigido usando L<sup>A</sup>T<sub>E</sub>X, mas seu uso é fortemente recomendado, pois ele facilita diversas etapas do trabalho e o resultado final é muito bom<sup>1</sup>. Este modelo é distribuído com uma “colinha” dos principais comandos L<sup>A</sup>T<sub>E</sub>X e inclui comentários explicativos para auxiliá-lo com ele, sendo composto dos arquivos de exemplo (`tese.tex`, `artigo.tex`, `apresentacao.tex` e `poster.tex`) e de arquivos auxiliares:

- Arquivos com o conteúdo do trabalho:
  - `conteudo/folhas-de-rosto.tex` (capa, dedicatória etc.)
  - `conteudo/resumo-abstract.tex`
  - `conteudo/capitulos.tex`, `conteudo/apendices.tex`, `conteudo/anexos.tex` e demais arquivos carregados por eles (`XX-*.tex`, `apendice-pseudocodigo.tex`, `anexo-faq.tex`)
  - `bibliografia.bib` (dados bibliográficos)
- Arquivos com as *packages* usadas e suas configurações (leia os comentários neles se quiser modificar algum aspecto do documento ou acrescentar alguma *package*):
  - `extras/basics.tex` (*packages* e configurações essenciais), `extras/fonts.tex` (definição das fontes do documento) e `extras/floats.tex` (configurações e melhorias para *floats*)
  - `extras/thesis-formatting.tex` (aparência: espaçamento, sumário etc.)
  - `extras/index.tex` (configuração do índice remissivo)
  - `extras/hyperlinks.tex` (configuração das referências cruzadas)
  - `extras/source-code.tex` (exibição de código-fonte e pseudocódigo)
  - `extras/utils.tex` (*packages* adicionais diversas)

---

<sup>1</sup>O uso de um sistema de controle de versões, como mercurial ([mercurial-scm.org](http://mercurial-scm.org)) ou git ([git-scm.com](http://git-scm.com)), também é altamente recomendado.

- extras/bibconfig.tex (configuração da bibliografia)
- Outros arquivos auxiliares (geralmente não precisam ser editados):
  - extras/imeusp-capas.sty (formatação da capa e demais páginas iniciais)
  - extras/imeusp-headers.sty (formatação dos cabeçalhos)
  - extras/lstpseudocode.sty (suporte a pseudocódigo com listings)
  - extras/annex.sty (permite adicionar anexos) e extras/appendixlabel.sty (melhora a lista de apêndices/anexos no sumário)
  - frufru.sty (divisões com ornamentos/florões, como mais abaixo)
  - extras/beamer\*.sty (layouts e cores para apresentações e posters)
  - extras/plainnat-ime.\* (estilo plainnat para bibliografias)
  - extras/alpha-ime.bst (estilo alpha para bibliografias com bibtex)
  - extras/natbib-ime.sty (tradução da package padrão natbib)
  - hyperxindy.xdy (configuração para xindy) e mkidxhead.ist (configuração para makeindex) — criados automaticamente
  - latexmkrc e Makefile (automatizam a geração do documento com os comandos latexmk e make respectivamente)



Para compilar o documento, basta executar o comando `latexmk` (ou `make`)<sup>2</sup>. Talvez seu editor ofereça uma opção de menu para compilar o documento, mas ele provavelmente depende do `latexmk` para isso. L<sup>A</sup>T<sub>E</sub>X gera diversos arquivos auxiliares durante a compilação que, em algumas raras situações, podem ficar inconsistentes (causando erros de compilação ou erros no PDF gerado, como referências faltando ou numeração de páginas incorreta no sumário). Nesse caso, é só usar o comando `latexmk -C` (ou `make clean`), que apaga todos os arquivos auxiliares gerados, e em seguida rodar `latexmk` (ou `make`) novamente.

## 2.1 Instalação do L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X é, na verdade, um conjunto de programas. Ao invés de procurar e baixar cada um deles, o mais comum é baixar um pacote com todos eles juntos. Há dois pacotes desse tipo disponíveis: MiK<sub>T</sub>E<sub>X</sub> ([miktex.org](http://miktex.org)) e T<sub>E</sub>XLive ([www.tug.org/texlive](http://www.tug.org/texlive)). Ambos funcionam em Linux, Windows e MacOS X. Em Linux, T<sub>E</sub>XLive costuma estar disponível para instalação junto com os demais opcionais do sistema. Em MacOS X, o mais popular é o Mac<sub>T</sub>E<sub>X</sub> ([www.tug.org/mactex/](http://www.tug.org/mactex/)), a versão do T<sub>E</sub>XLive para MacOS X. Em Windows, o mais comumente usado é o MiK<sub>T</sub>E<sub>X</sub>.

<sup>2</sup>Você também pode usar `latexmk poster`, `make apresentacao` etc.



Por padrão, eles não instalam tudo que está disponível, mas sim apenas os componentes mais usados, e oferecem um gestor de pacotes que permite adicionar outros. Embora uma instalação completa do  $\LaTeX$  seja relativamente grande (perto de 5GB), em geral vale a pena instalar a maior parte dos pacotes. Se você preferir uma instalação mais “enxuta”, não deixe de incluir todos os pacotes necessários para este modelo, como indicado no arquivo README.md.

Também é muito importante ter o `latexmk` (ou o `make`). No Linux, a instalação é similar à de outros programas. No MacOS X e no Windows, `latexmk` pode ser instalado pelo gestor de pacotes do  $\text{MiKTeX}$  ou  $\text{T\TeX Live}$ . Observe que ele depende da linguagem `perl`, que precisa ser instalada à parte no Windows ([www.perl.org/get.html](http://www.perl.org/get.html)).

## 2.2 Documentação sobre $\LaTeX$

Existem diversos bons livros sobre  $\LaTeX$  (embora em geral um tanto antigos), dos quais destacamos dois:

1. A quarta edição de “A Guide to  $\LaTeX$ ”, de Helmut Kopka e Patrick W. Daly (publicada em 2003), além de uma ótima introdução, aborda vários tópicos relativamente avançados e úteis<sup>3</sup>.
2. A segunda edição de “The  $\LaTeX$  Companion” (publicada em 2004) é um livro quase obrigatório, pois discute em detalhes praticamente todos os recursos e *packages* importantes de  $\LaTeX$ , servindo tanto para o aprendizado quanto como material de referência.

Além de livros, há muito material sobre  $\LaTeX$  na Internet, mas também há muita informação obsoleta. Em particular, você pode ignorar explicações sobre como converter arquivos no formato DVI gerados por  $\LaTeX$  em PDF: As versões atualmente recomendadas de  $\LaTeX$  (cf. Seção 3.11) geram arquivos PDF diretamente. Quanto a imagens, os formatos de arquivo PS/EPS (PostScript e Encapsulated PostScript) não são adequados para essas novas versões de  $\LaTeX$ ; elas trabalham com arquivos de imagem nos formatos PDF, PNG e JPEG. Finalmente, recursos gráficos normalmente não usam mais *packages* como `pstricks`, `eepic` ou outras tradicionalmente citadas; ao invés disso, `PGF/TikZ` é a ferramenta mais comum.

Dentre o material online, o conteúdo em [overleaf.com/learn](http://overleaf.com/learn) é excelente, incluindo um rápido tutorial (de escopo similar ao Capítulo 3 deste modelo) e várias páginas sobre como utilizar recursos específicos. Um tutorial bastante abrangente e detalhado está disponível em [tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf](http://tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf) (há outros, como [tug.org/tutorials/tugindia](http://tug.org/tutorials/tugindia), [www.maths.tcd.ie/~dwilkins/LaTeXPrimer/GSWLaTeX.pdf](http://www.maths.tcd.ie/~dwilkins/LaTeXPrimer/GSWLaTeX.pdf) e [www.andy-roberts.net/writing/latex](http://www.andy-roberts.net/writing/latex)). Em português, você pode consultar [polignu.org/sites/polignu.org/files/latex/latex-fflch.pdf](http://polignu.org/sites/polignu.org/files/latex/latex-fflch.pdf) e [git.febrace.org.br/material-latex/material-latex](http://git.febrace.org.br/material-latex/material-latex) (este precisa ser baixado e compilado). O sítio [tex.stackexchange.com](http://tex.stackexchange.com) é um fórum de perguntas e respostas sobre  $\LaTeX$  muito útil, pois os principais desenvolvedores do sistema participam das

---

<sup>3</sup>Um dos autores disponibiliza uma versão não-final em [www2.mps.mpg.de/homes/daly/GTL/gtl\\_20030512.pdf](http://www2.mps.mpg.de/homes/daly/GTL/gtl_20030512.pdf).

discussões, e o sítio [texfaq.org](http://texfaq.org) é bastante abrangente e atualizado. O canal [youtube.com/c/anteroneves](https://www.youtube.com/c/anteroneves) tem vários vídeos instrutivos em português.

Como dito anteriormente, L<sup>A</sup>T<sub>E</sub>X é, na verdade, um conjunto de programas e, normalmente, instalamos pacotes pré-prontos com todos eles. Esses pacotes (em geral, T<sub>E</sub>XLive e MiK<sub>T</sub><sub>E</sub>X) contém também a documentação das *packages* incluídas: Basta digitar `texdoc nome-da-package` (T<sub>E</sub>XLive) ou `mtxhelp nome-da-package` (MiK<sub>T</sub><sub>E</sub>X) para ter acesso à documentação correspondente. `texdoc/mtxhelp` incluem também alguns tutoriais e textos introdutórios, como “The Not So Short Introduction to L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>” (`texdoc lshort-eng`; há uma versão em português, mas não está em dia com o original), “Getting up and running with A<sub>M</sub>S-L<sup>A</sup>T<sub>E</sub>X” (`texdoc amshelp`) e “A Simplified Introduction to L<sup>A</sup>T<sub>E</sub>X” (`texdoc simplified-intro`). Versões recentes do L<sup>A</sup>T<sub>E</sub>X incluem também o “L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> via exemplos” (`texdoc latex-via-exemplos`), em português. O sítio [texdoc.net](http://texdoc.net) oferece acesso online a essa mesma documentação de maneira organizada e o sítio [ctan.org](http://ctan.org) é o repositório semi-oficial das *packages* L<sup>A</sup>T<sub>E</sub>X e sua documentação.

A documentação de referência de L<sup>A</sup>T<sub>E</sub>X pode ser acessada com `texdoc latex2e`. Minúcias sobre seu funcionamento interno estão descritas em `texdoc source2e` e, sobre as classes padrão (`article`, `book` etc.), em `texdoc classes`. Você normalmente não vai usar esses documentos, mas eles podem servir para esclarecer algum detalhe. `texdoc clsguide` é um guia para a criação de novas classes e *packages*, e `texdoc macros2e` é uma lista de comandos especialmente úteis para isso. Finalmente, `texdoc fntguide` explica como funciona a gestão de fontes de L<sup>A</sup>T<sub>E</sub>X (mas note que LuaL<sup>A</sup>T<sub>E</sub>X e XeL<sup>A</sup>T<sub>E</sub>X usam outro mecanismo; veja `texdoc fontspec`). Você pode ver exemplos de fontes disponíveis para L<sup>A</sup>T<sub>E</sub>X em [tug.org/FontCatalogue](http://tug.org/FontCatalogue).

Quando você se tornar um usuário avançado, pode se interessar em conhecer melhor a linguagem T<sub>E</sub>X, que está na base do L<sup>A</sup>T<sub>E</sub>X. “The T<sub>E</sub>X book”, de Donald Knuth (o criador do T<sub>E</sub>X), é amplamente recomendado, mas há três livros completos a respeito que são instalados com L<sup>A</sup>T<sub>E</sub>X: “A gentle introduction to T<sub>E</sub>X” (`texdoc gentle`), “T<sub>E</sub>X for the impatient” (`texdoc impatient`) e “T<sub>E</sub>X by topic” (`texdoc texbytopic`).

## Chapter 3

# Do zero ao mínimo com $\text{\LaTeX}$

Preparar um texto para impressão envolve duas coisas:

**Escrever:** digitar, recortar/colar trechos, revisar etc.

**Formatar:** definir o tamanho da fonte, o espaçamento entre parágrafos etc.

Hoje é comum fazer essas duas coisas ao mesmo tempo, graças à visualização imediata que o computador oferece. No entanto, imagine como era o processo de produção de um livro nos anos 1970: o autor escrevia seu texto em uma máquina de escrever e enviava esse material para o editor, que era responsável pela tarefa de formatá-lo para impressão. O autor muitas vezes inseria anotações para o editor explicando coisas como “este parágrafo é uma citação”, e o editor criava algum mecanismo visual para representar isso.

Não é de se surpreender que, com o surgimento do microcomputador, os primeiros programas para criação de textos seguissem um funcionamento similar: o autor digitava e editava seu texto sem formatá-lo visualmente, apenas inserindo alguns comandos correspondentes a aspectos da formatação que ele depois revisava na versão impressa.  $\text{\LaTeX}$  é uma ferramenta baseada nesse processo: você prepara seu texto no editor de sua preferência, insere comandos no texto que indicam a estrutura do documento e o processa com o  $\text{\LaTeX}$ , que gera um arquivo PDF formatado. Embora seja um estilo “antigo” de trabalhar, ele é muito eficiente em vários casos. Ou seja, dependendo da situação, pode ser mais adequado trabalhar fazendo tudo ao mesmo tempo ou dividindo o trabalho nessas duas fases. De maneira geral:

- Se você precisa criar páginas diferentes entre si com *layout* definido manualmente, é melhor usar uma ferramenta que permita trabalhar visualmente, como LibreOffice Writer, MS-Word, Google Docs etc.;
- Se você precisa fazer um documento relativamente longo com estrutura regular (capítulos, seções etc.), é melhor usar ferramentas que formalizam essa estrutura (como  $\text{\LaTeX}$ ) ao invés de ferramentas visuais;
- Se você precisa fazer um documento envolvendo referências cruzadas, bibliografia relativamente extensa ou fórmulas matemáticas, é difícil encontrar outra ferramenta tão eficiente quanto  $\text{\LaTeX}$ ;

- Se você precisa criar um documento simples, ambas as abordagens funcionam bem; cada um escolhe esta ou aquela em função da familiaridade com as ferramentas;
- Se você quer que a qualidade tipográfica do resultado seja realmente excelente, é necessário usar uma ferramenta profissional, como  $\text{\LaTeX}$ , Scribus, Adobe InDesign ou outras; processadores de texto convencionais não oferecem o mesmo nível de qualidade dessas ferramentas.

### 3.1 Visão Geral

Com  $\text{\LaTeX}$ , você prepara o texto (incluindo as indicações de estrutura) em um editor de textos qualquer, salva como arquivo de texto puro (“.txt”, mas é comum usar a extensão “.tex” ao invés de “.txt”) e processa esse arquivo com o comando “pdflatex” (“compila” o documento) para obter o PDF correspondente. Qualquer editor capaz de salvar arquivos em formato texto puro, como o bloco de notas do windows, vim, emacs etc. pode ser usado. Programas como LibreOffice Writer, MS-Word etc. também funcionam, mas possivelmente vão gerar dores de cabeça porque vão tentar formatar algumas coisas automaticamente (e de maneira incompatível com  $\text{\LaTeX}$ ).

Se você preferir, existem editores projetados especificamente para trabalhar com  $\text{\LaTeX}$ ; eles em geral utilizam cores para distinguir o texto dos comandos de formatação, automatizam o processo de compilação do documento e oferecem outras comodidades. Os mais comumente usados são  $\text{\TeX}$ maker,  $\text{\TeX}$ studio e  $\text{\TeX}$ works; os três são software livre e funcionam em Windows, MacOS e Linux.  $\text{\TeX}$ nicCenter é outra opção livre, mas funciona apenas em Windows. Os editores atom ([atom.io](https://atom.io)) e Visual Studio Code ([code.visualstudio.com](https://code.visualstudio.com)) têm interfaces às vezes peculiares para não programadores, mas em conjunto com *packages* adicionais (atom-latex, latex-document-outline, grammar-token-limit e preview-inline para atom e LaTeX Workshop para vscode), são uma boa opção (observe que as *packages* mencionadas são do editor, não do  $\text{\LaTeX}$ ). O mesmo vale para o editor emacs ([www.gnu.org/software/emacs](https://www.gnu.org/software/emacs)) e sua package AU $\text{\TeX}$ . Ainda outra possibilidade são os editores *online*, como overleaf ([www.overleaf.com](https://www.overleaf.com)), authorea ([www.authorea.com](https://www.authorea.com)) e papeeria ([papeeria.com](https://papeeria.com)).

Um documento  $\text{\LaTeX}$  é dividido em duas partes: o *preâmbulo*, onde você coloca comandos de configuração para o documento, e o *corpo* do documento em si, que contém o texto propriamente dito. O preâmbulo é onde você define as características do resultado tipográfico esperado para o documento como um todo: tipo e tamanho da fonte a usar, posição dos títulos e subtítulos na página etc. O corpo, por sua vez, consiste no texto e em alguns comandos indicativos da estrutura.

Dado que configurar o preâmbulo é um tanto complexo e que mesmo no corpo do texto às vezes há comandos especiais (para a geração da bibliografia ou tabelas, por exemplo), usar algum documento existente como base para criar seu texto em geral é uma boa ideia. O IME/USP oferece um conjunto de modelos adequados para teses/dissertações, artigos, apresentações e pôsteres ([gitlab.com/ccsl-usp/modelo-latex](https://gitlab.com/ccsl-usp/modelo-latex)) que pode ser adaptado para outros usos e outras instituições. Há também uma família de modelos ([www.abntex.net.br](https://www.abntex.net.br)) que procura seguir as normas da ABNT para diversos tipos de documentos científicos, e algumas publicações científicas fornecem modelos de acordo com suas diretrizes.

## 3.2 Estrutura de um Documento $\LaTeX$

O preâmbulo  $\LaTeX$  começa com a definição da *classe* a ser utilizada, que determina boa parte da configuração do documento. As principais classes são book, report e article; você pode saber mais sobre elas (e outras) em qualquer texto introdutório sobre  $\LaTeX$  na Internet (veja a Seção 2.2). book e report são as mais adequadas para a escrita de teses ou dissertações acadêmicas. A seguir, são carregadas várias *packages* (“*plugins*”) que acrescentam funcionalidades ou modificam as classes padrão. Qualquer documento  $\LaTeX$  utiliza várias delas e é comum que revistas científicas utilizem *packages* próprias que pré-definem a formatação esperada para os artigos. A classe é definida com o comando `\documentclass{nome-da-classe}`; *packages* são carregadas com o comando `\usepackage{nome-da-package}`. Classes e *packages* podem receber opções adicionais entre colchetes (`\usepackage[opção1,opção2...]{nome-da-package}`); a documentação de cada *package* e classe (veja a Seção 2.2) detalha as opções disponíveis.

$\LaTeX$  ignora quebras de linha e trata sequências de vários espaços como se fossem apenas um. Isso significa que você pode usar quebras de linha e espaços no texto que está digitando como “dicas visuais” da estrutura do texto durante a edição. É muito comum fazer isso com listas de itens, por exemplo. Uma ou mais linhas em branco sinalizam o fim de um parágrafo e o início de outro. O caractere “%” indica que o restante da linha é um comentário, ou seja, um trecho de texto que não tem nenhum efeito sobre o resultado final do documento. Comentários podem ser usados como lembrete sobre alguma decisão, para indicar um parágrafo que ainda precisa de revisão etc. Por conta desse significado especial, para inserir um caractere % “normal” no texto é preciso digitar “\%”.

Como mencionado anteriormente,  $\LaTeX$  divide o trabalho de produção de um texto entre a preparação do conteúdo e a definição da forma de apresentação. Assim, os comandos usados durante a produção do conteúdo procuram expressar o *significado* de cada elemento, e não sua aparência. Por exemplo, para realçar uma palavra é comum usar texto *em itálico*; embora exista um comando especificamente para gerar textos em itálico em  $\LaTeX$ , o recomendado é que se utilize o comando `\emph` (“ênfático”), pois em alguns casos pode ser melhor utilizar **negrito**, `VERSALETE` ou outro mecanismo para dar ênfase a uma palavra. Essa é uma orientação geral para a escrita de textos com  $\LaTeX$ : procure definir a estrutura, não a aparência.

Um exemplo de documento  $\LaTeX$  simples:

```
% O documento começa com o preâmbulo
% Vamos usar a classe "book" com fonte no tamanho 11pt
\documentclass[11pt]{book}
% Vamos usar caracteres acentuados
\usepackage[utf8]{inputenc}
% Vamos escrever em português do Brasil
\usepackage[brazil]{babel}
% Estas linhas não imprimem nada, apenas definem
% as informações que serão usadas por "\maketitle"
\author{Fulano de Tal}
\title{Começando a usar o \LaTeX{}}
```

```
% Finaliza o preâmbulo e inicia o conteúdo:
\begin{document}
% Cria uma página de título com os dados definidos acima
\maketitle
% Capítulos, seções etc. são numerados automaticamente
\chapter{Cheguei!}
Oi, Galera!
% É preciso sinalizar o final do documento
\end{document}
```

Esse exemplo mostra como definir o nome de um capítulo. Existem também os comandos `\section`, `\subsection`, `\subsubsection` e `\paragraph` (a classe `book` inclui também `\part`, um nível acima de `\chapter`). Usar o nome do comando seguido de um asterisco (`\chapter*` etc.) faz o capítulo/seção não ser numerado nem incluído no sumário (nem considerado na contagem de capítulos, seções etc.).

### 3.3 Executando L<sup>A</sup>T<sub>E</sub>X e Comandos Auxiliares

Depois de escrever o arquivo `.tex`, é preciso *compilá-lo*, ou seja, processá-lo para gerar o pdf desejado. Isso envolve executar, além do próprio L<sup>A</sup>T<sub>E</sub>X (veja a Seção 3.11), alguns programas auxiliares (em geral, `biber` ou `bibtex` e `makeindex`). Nesse processo, L<sup>A</sup>T<sub>E</sub>X quase sempre precisa ser executado três ou mais vezes antes de gerar o pdf final<sup>1</sup>. Por conta dessa complexidade, é comum utilizar alguma ferramenta para automatizar o processamento. Existem diversas opções, mas a mais comum é o `latexmk`, que é capaz de identificar automaticamente os passos necessários para a geração do documento, executando os programas na ordem correta quantas vezes forem necessárias<sup>2</sup>. Assim, embora seja possível gerar o pdf executando apenas `pdflatex nome-do-arquivo.tex`, acostume-se a compilar o documento sempre com `latexmk nome-do-arquivo.tex`. Note que editores especializados em L<sup>A</sup>T<sub>E</sub>X costumam ter uma opção de menu para a compilação do documento; muitas vezes essa opção simplesmente aciona `latexmk`.

### 3.4 Mais sobre Estrutura

Para criar listas de itens, você pode fazer<sup>3</sup>:

```
\begin{itemize}
  \item Primeiro item
  \item Segundo item
  \item Terceiro item
\end{itemize}
```

<sup>1</sup>A cada vez, ele gera uma nova versão intermediária do arquivo pdf, mas essas versões têm defeitos, como citações e referências cruzadas incorretas ou sumário inexistente.

<sup>2</sup>É possível personalizar o comportamento de `latexmk` com o arquivo de configuração `latexmkrc`.

<sup>3</sup>Observe o uso de espaços no início das linhas com `\item` para deixar a estrutura visualmente mais clara durante a edição.

Além de “itemize”, há também “enumerate” (auto-explicativo) e “description”:

```
\begin{description}
  \item[O primeiro item] é o primeiro;
  \item[O segundo item] é o segundo;
  \item[O terceiro item] é o terceiro.
\end{description}
```

Citações curtas normalmente são incluídas no fluxo normal do texto e colocadas entre aspas; para citações mais longas, use `\begin{quote}` ou `\begin{quotation}` (este último é mais adequado para citações com vários parágrafos). Para poesia, use `verse` (estrofes são separadas por uma linha em branco e versos são separados por `\*`. O asterisco é opcional; ele instrui  $\LaTeX$  a manter as linhas na mesma página). A package `csquotes` acrescenta recursos sofisticados para citações.

Para inserir uma nota de rodapé, use o comando `\footnote{texto da nota}`. Um espaço não-separável é indicado pelo caractere til (“~”) e é possível forçar uma quebra de linha com “\”. Aspas tipográficas (“” e ‘’) são inseridas com ```''` e ``'`'`. Você pode consultar a lista completa de símbolos com `texdoc symbols-a4` ou em [www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf](http://www.ctan.org/tex-archive/info/symbols/comprehensive/symbols-a4.pdf). Uma outra maneira de encontrar símbolos é usar este sítio: [detexify.kirelabs.org/classify.html](http://detexify.kirelabs.org/classify.html).

## 3.5 Referências Cruzadas e Floats

É comum que um trecho do texto faça referência a outro trecho (“como discutimos no Capítulo X...”). Isso pode ser feito diretamente, mas se você reorganizar o documento ou acrescentar seções, a numeração pode mudar. Para evitar esse problema, você pode gerar essas referências automaticamente com o par de comandos `\label{nome-sugestivo}` e `\ref{nome-sugestivo}` (para o número da seção/capítulo) ou `\pageref{nome-sugestivo}` (para o número da página).

Esse mecanismo também é muito útil para figuras e tabelas. É claro que o ideal seria que tabelas e figuras sempre aparecessem junto ao texto a que se referem. No entanto, isso é impossível por conta da divisão do texto em páginas. Em  $\LaTeX$ , é melhor incluir figuras e tabelas como *floats* (localização flexível) usando `\begin{figure}` e `\begin{table}` e deixar o programa procurar o “melhor” lugar para colocá-las. A figura/tabela em si é definida com `\includegraphics` ou `\begin{tabular}`, e em geral é uma boa ideia acrescentar uma legenda com `\caption`. Finalmente, dentro da legenda, é possível inserir um `\label` para que se possa fazer referência à figura/tabela no texto (com os comandos `\ref` e `\pageref`)<sup>4</sup>.

$\LaTeX$  garante que a sequência das figuras e a sequência das tabelas sejam respeitadas (a Figura 6 nunca aparece depois da Figura 7). No entanto, isso *não* se aplica a *floats* de tipos diferentes, ou seja, se você definiu a Figura 5, a Tabela 3 e a Figura 6, elas podem aparecer no documento na ordem “Figura 5, Tabela 3, Figura 6”, “Figura 5, Figura 6, Tabela 3” ou “Tabela 3, Figura 5, Figura 6”.

<sup>4</sup>Em alguns casos, é possível colocar o `\label` de uma figura ou tabela fora do comando `\caption` mas, como em muitos casos isso gera problemas, é um bom hábito sempre colocá-lo dentro dele.



## 3.6 Fórmulas Matemáticas

A diagramação de fórmulas matemáticas tem regras específicas; assim, para criar fórmulas em  $\text{\LaTeX}$ , é preciso usar um comando para iniciar o modo matemático. Isso pode ser feito de duas formas:

- Pequenas fórmulas no meio do texto ( $E = mc^2$ ) são inseridas com  $\$fórmula\$$  (e, portanto, para inserir um caractere \$ normal no texto, é preciso usar  $\backslash\$$ ).
- Fórmulas mais longas ou que devem aparecer em um parágrafo separado são inseridas com  $\backslash[fórmula\]$  (ou  $\backslashbegin{displaymath}$ ).

No modo matemático, letras são interpretadas como variáveis e espaços em branco são ignorados ( $\text{\LaTeX}$  usa o contexto da fórmula para definir o espaçamento). Para inserir um espaço explicitamente, use  $\backslashquad$  ou  $\backslashenspace$ . Para inserir texto “normal” em uma fórmula matemática, use  $\backslashtext{texto}$  (para texto de fato) ou  $\backslashmathit{texto}$  (para nomes de variáveis ou funções com mais de uma letra). Pode ser necessário deixar um espaço no início do texto para evitar que ele fique colado com o caractere matemático que o antecede.

Usando  $\backslashbegin{equation}$ , a fórmula recebe um número (que aparece à direita) ao qual você pode se referir no texto usando os comandos “ $\backslashref$ ” e “ $\backslasheqref$ ” (“conforme vimos na equação  $\backslashref{eq:bhaskara}$ ...” ou “de acordo com  $\backslasheqref{eq:bhaskara}$ ...”).  $\backslashbegin{equation*}$  (incluindo o  $*$ ) elimina o número e é, portanto, equivalente a  $\backslashbegin{displaymath}$ . Há outros comandos similares, como  $\text{align}$ ,  $\text{multline}$  e  $\text{gather}$ , definidos e documentados na package  $\text{amsmath}$ , e todos têm a variante com “ $*$ ”.

## 3.7 Referências Bibliográficas e Bibliografia

A geração de bibliografias no  $\text{\LaTeX}$  é feita através da package  $\text{biblatex}$  e do programa auxiliar  $\text{biber}$ <sup>5</sup> e envolve três passos:

1. A criação de um banco de dados, no formato “.bib”, das obras de interesse. Esse banco de dados pode incluir obras que não vão ser de fato referenciadas no documento final. Isso significa que você pode criar um único banco de dados e utilizá-lo em todos seus documentos<sup>6</sup>.
2. A inserção de referências às obras ao longo do texto, usando diferentes comandos dependendo do caso:  $\backslashcite$ ,  $\backslashcitet$ ,  $\backslashcitep$  etc. Como já mencionado, esses comandos estão descritos na documentação da package  $\text{natbib}$  (DALY, 2010).
3. A escolha do estilo bibliográfico (usando as opções da package  $\text{biblatex}$ ) que formata as citações ao longo do texto e gera a bibliografia automaticamente através do comando  $\backslashprintbibliography$ . Normalmente, apenas as obras efetivamente citadas são incluídas na lista de referências, mas é possível forçar a inclusão de uma obra sem citá-la explicitamente com o comando  $\backslashnocite$ .

<sup>5</sup>Antigamente, usava-se a package  $\text{natbib}$  e o comando auxiliar  $\text{bibtex}$ . O funcionamento geral dos dois mecanismos é similar e o formato do banco de dados de ambos é o mesmo.

<sup>6</sup>É comum criar bancos de dados desse tipo separados por assunto, mas isso não é necessário.



O banco de dados é um arquivo de texto contendo uma *entrada* para cada item da bibliografia e, em cada entrada, uma série de *campos* com os dados (título, autor etc.). A entrada inclui também uma *chave*, que é usada para inserir as citações no texto. Há vários tipos de entrada (para artigos, livros, sítios web etc.) e, para cada tipo, uma lista de campos possíveis (considere que periódicos normalmente incluem o número do volume, mas teses não). O exemplo abaixo é um livro cuja chave é “dissertjourney”; ele pode ser citado com o comando `\cite{dissertjourney}`:

```
@book{dissertjourney,
  author    = {Carol M. Roberts},
  title     = {The Dissertation Journey},
  publisher = {Corwin},
  year      = 2010,
  edition   = 2,
  location  = {Thousand Oaks, CA},
}
```

Observe que existem dois formatos comumente usados para escrever títulos de artigos, livros etc:

**Title case:** Substantivos, adjetivos e verbos (além de nomes próprios e siglas) são escritos com a primeira letra maiúscula (“Um Exemplo de Título no Estilo Title Case”). Em geral, a regra não se aplica ao título de artigos ou capítulos de livro, apenas aos livros dos quais eles fazem parte;

**Sentence case:** O título é escrito como qualquer outra frase (“Um título só tem maiúsculas em abreviaturas, como ABNT, ou nomes próprios”).

Cada estilo de bibliografia utiliza um desses formatos e, portanto, é desejável que o banco de dados funcione corretamente com ambos. No entanto, nem sempre é claro quais palavras devem ser iniciadas com letra maiúscula ao usar *title case* e, por conta disso, não há um sistema automático em  $\text{\LaTeX}$  para adaptar títulos a ele. Sendo assim, como fazer um banco de dados bibliográfico capaz de funcionar com os dois formatos? A solução é sempre inserir os títulos dos itens no banco de dados seguindo o formato *title case*. Se o estilo utiliza esse formato, o título é reproduzido na bibliografia como digitado no banco de dados. Se o estilo usa *sentence case*, o texto (exceto a primeira letra) é convertido para letras minúsculas. Para evitar que isso afete siglas e nomes próprios, basta colocá-los entre chaves (“Automated Application-Level Checkpointing of {MPI} Programs”).

Finalmente, os campos `author` e `publisher` podem incluir uma lista de nomes separados por `and`; `biblatex` reconhece que cada nome é composto por nome e sobrenome, às vezes com partículas como “de”, “dos” ou “von” e, dependendo do estilo bibliográfico, pode abreviar nomes, mudar sobrenomes para caixa alta etc. Isso evidentemente não funciona quando o autor é, na verdade, uma instituição; nesses casos, basta colocar o nome inteiro da instituição entre chaves (“{Universidade de São Paulo — Sistema Integrado de Bibliotecas}”) para que `biblatex` não faça alterações desse tipo. Se o nome é longo, pode ser interessante definir o campo `shortauthor`.

A fonte mais detalhada de informações sobre o banco de dados é a documentação da `package biblatex` (LEHMAN *et al.*, 2018, em especial as seções 2.1.1 e 2.2.2), mas o material ali

é um tanto denso. Há muito material introdutório ao formato “.bib” e ao bibtex disponível *online*, e você pode se inspirar em exemplos para criar seu banco de dados bibliográfico. Além disso, ferramentas como Zotero ou Mendeley (o uso de uma delas é altamente recomendado!) podem exportar para o formato .bib. Observe que biblatex oferece recursos bastante sofisticados para o tratamento de referências e bibliografias. Se você precisar de alguma funcionalidade especial, consulte a documentação do pacote ou a Internet; é quase certeza que biblatex oferece uma solução.

## 3.8 Imagens, Ilustrações, Diagramas e Gráficos

Podemos classificar imagens em quatro categorias:

1. Imagens fotográficas ou escaneadas, que consistem em um conjunto de *pixels* coloridos sem organização previsível.
2. Ilustrações, que consistem em curvas e figuras geométricas que formam uma imagem completa, como um objeto ou uma paisagem. Apesar de lidarem com abstrações geométricas ao invés de meros *pixels*, elas ainda são desenhadas de forma totalmente manual em programas como Inkscape ou CorelDraw!.
3. Diagramas, que são ilustrações estruturadas, como fluxogramas, grafos ou diagramas UML, criadas com ferramentas como Draw.io, LibreOffice Draw ou Microsoft Visio. Graças à sua estrutura intrínseca, os programas podem automatizar, ao menos parcialmente, o trabalho de posicionar e alinhar cada elemento.
4. Gráficos de dados, como gráficos de pizza ou de barras. A geração desses gráficos, em geral, é quase totalmente automatizada por ferramentas como Gnuplot, R, LibreOffice Calc ou Microsoft Excel.

Em L<sup>A</sup>T<sub>E</sub>X, é possível importar imagens fotográficas nos formatos PNG e JPG e imagens dos demais tipos no formato PDF. Além disso, L<sup>A</sup>T<sub>E</sub>X tem recursos para criar ilustrações, diagramas e gráficos diretamente, mas usá-los em geral não é trivial. Em particular, a package tikz oferece bons recursos para a criação de ilustrações e diagramas (incluindo funções pré-prontas para formas geométricas, grafos, matrizes etc.) e é fácil usá-la para traçar linhas ou curvas simples. Você também pode criar gráficos de dados ou de funções matemáticas com a package pgfplots. Gnuplot, com o *driver* lua tikz<sup>7</sup>, e matplotlib, com o *backend* PGF<sup>8</sup>, são capazes de exportar gráficos de dados na forma de comandos para tikz (garantindo maior consistência visual entre o texto principal e o gráfico), e o programa Asymptote tem excelente integração com L<sup>A</sup>T<sub>E</sub>X.

## 3.9 Formatação Manual

Às vezes é preciso inserir formatação de forma manual; os comandos mais importantes são: \emph (texto *ênfatisado*, em geral itálico), \texttt (texto *teletype*, imitando um terminal de texto ou uma impressora), \textit (*itálico*), \textbf (**negrito**), \textsf (fonte

<sup>7</sup>[www.gnuplot.info/docs\\_5.2/Gnuplot\\_5.2.pdf#section\\*.516](http://www.gnuplot.info/docs_5.2/Gnuplot_5.2.pdf#section*.516)

<sup>8</sup>[matplotlib.org/users/pgf.html](http://matplotlib.org/users/pgf.html)

sem serifa), `\textsc` (texto VERSALETE — nem todas as fontes oferecem essa possibilidade), `\normalsize` (tamanho normal), `\small` (tamanho reduzido), `\footnotesize` (ainda menor), `\scriptsize` (ainda menor), `\tiny` (ainda menor), `\large` (tamanho aumentado), `\Large` (ainda maior), `\LARGE` (ainda maior), `\Huge` (ainda maior), `\vspace{\baselineskip}` (deixa uma linha em branco), `\begin{center}` (centraliza parágrafos), `\begin{flushleft}` (alinha parágrafos à esquerda), `\begin{flushright}` (alinha parágrafos à direita)<sup>9</sup>, `\hyphenation` (permite “ensinar”  $\text{\LaTeX}$  como hifenizar uma lista de palavras; note que, em geral, a hifenização automática de  $\text{\LaTeX}$  é excelente), `\-` (sugere uma possível hifenização localizada), `\leftskip=1cm` (aumenta a margem esquerda), `\rightskip=1cm` (aumenta a margem direita), `\linebreak[0–4]` (sugere uma quebra de linha; o número indica quão forte é a sugestão, ou seja, 4 faz a quebra obrigatória; se o parágrafo é justificado, a linha quebrada também é justificada), `\newline` ou `\\` (força uma quebra de linha; a linha *não* é justificada nesse caso), `\pagebreak[0–4]` (sugere uma quebra de página; como `\linebreak`, o número indica quão forte é a sugestão; o texto da página é espalhado verticalmente de maneira a fazer a última linha alinhada com o final das demais páginas) e `\newpage` (força uma quebra de página; o final da página *não* é alinhado com o final das demais páginas nesse caso).

Mas, como discutido na Seção 3.2, não é recomendável usar esses comandos ao longo do texto: o ideal em  $\text{\LaTeX}$  é expressar o significado de cada elemento, não a sua forma de apresentação, pois isso permite que você faça alterações na formatação com mais facilidade. Assim, quando os recursos pré-definidos do  $\text{\LaTeX}$  (`\itemize`, `\chapter` etc.) não forem suficientes, o mais adequado é definir comandos novos, em geral usando os comandos de formatação mencionados acima. Esse é um tópico avançado, mas você pode consultar o início do arquivo  $\text{\LaTeX}$  deste capítulo para alguns exemplos simples.

## 3.10 Detalhes da Linguagem

Há quatro estilos típicos de comandos  $\text{\LaTeX}$ :

- Comandos que se referem a um parâmetro; por exemplo, `\emph{um texto}` significa “escreva a frase ‘um texto’ com ênfase” (em geral, *itálico*). As chaves delimitam o início e o final do escopo sobre o qual o comando tem efeito. Aqui entram também comandos como `\title` e `\author`, que não escrevem nada diretamente mas definem o título e autoria do documento (essa informação é usada, por exemplo, por `\maketitle`).
- Comandos que se referem a um parâmetro que é um bloco grande de texto, possivelmente vários parágrafos; por exemplo, `\begin{center} um texto \end{center}` faz “um texto” (que podem ser vários parágrafos) ser centralizado.
- Comandos que ativam alguma opção; por exemplo, `\itshape` significa “ative o modo *itálico*”. Nesse caso, o texto vai ser impresso em *itálico* até outro comando selecionar outro estilo de fonte. Se o comando for inserido dentro de um bloco delimitado por chaves, ele “perde o efeito” após o caractere de fecha-chaves (exemplo: “`\{ \itshape \}` Fulano de Tal” é meu nome” será impresso como “*Fulano de Tal* é meu nome”). Você normalmente não vai utilizar esse estilo de comando, mas ele é útil em alguns casos.

---

<sup>9</sup>É altamente recomendável carregar a package `ragged2e` (já incluída neste modelo) e utilizar `\Center`, `\FlushLeft` e `\FlushRight` ao invés de `\center`, `\flushleft` e `\flushright`.

- Comandos que fazem o programa escrever algo específico; por exemplo, em várias classes padrão o comando `\maketitle` gera uma página de título com o nome do trabalho, autor etc.

Nos dois últimos, não é preciso usar chaves após o comando. Ainda assim, as chaves podem ser colocadas e muitas vezes isso é bom: sem elas,  $\text{\LaTeX}$  entende que o caractere espaço que se segue a esses comandos serve apenas como separador em relação ao que vem a seguir. Por conta disso, ele ignora esse espaço. Quando isso não é o que se deseja, a solução é usar as chaves: `\itshape{}`. Vale observar que alguns comandos aceitam mais de um parâmetro, às vezes entre chaves, às vezes entre colchetes. Você pode descobrir a sintaxe correta para cada caso lendo a documentação de cada comando.

### 3.11 Versões do $\text{\LaTeX}$

Assim como há packages para o  $\text{\LaTeX}$ , o próprio  $\text{\LaTeX}$  é, na verdade, um conjunto de extensões para o programa  $\text{\TeX}$ . Assim, se você encontrar referências a “ $\text{\TeX}$ ” ou a “plain  $\text{\TeX}$ ”, basta saber que esse é o sistema que funciona “por baixo” do  $\text{\LaTeX}$ .

$\text{\LaTeX}$  é um sistema em evolução (desde os anos 80!). Uma das consequências disso é que há, na verdade, quatro versões diferentes dele:

1.  $\text{\LaTeX}$  “tradicional”, que gera arquivos em formato DVI que, por sua vez, precisam ser convertidos para o formato PDF. Essa versão não é capaz de usar as fontes instaladas no sistema; ela só pode usar fontes adaptadas para uso com o  $\text{\LaTeX}$ . Hoje em dia não há boas razões para usar essa versão.
2.  $\text{\pdfLaTeX}$ , que gera arquivos PDF e dá suporte a alguns recursos avançados de tipografia adicionais. É a versão mais usada hoje em dia, embora também só possa usar as fontes adaptadas para uso com o  $\text{\LaTeX}$ .
3.  $\text{\XeLaTeX}$  que, além dos recursos do  $\text{\pdfLaTeX}$ , opera internamente em UTF-8 (ou seja, funciona melhor com múltiplas línguas) e pode funcionar não só com as fontes adaptadas para o  $\text{\LaTeX}$  como também com as fontes instaladas no sistema.
4.  $\text{\LuaLaTeX}$ , que oferece os mesmos recursos que o  $\text{\XeLaTeX}$  e também pode ser estendido internamente com mais facilidade (através da linguagem de programação Lua).

Todas essas versões são instaladas quando você instala o  $\text{\LaTeX}$  na sua máquina.  $\text{\XeLaTeX}$  e  $\text{\LuaLaTeX}$  são as duas propostas da comunidade para o futuro novo padrão do sistema, mas você não tem nada a perder se escolher a “errada”, pois para todos os efeitos práticos elas são equivalentes. Em geral, se você pretende escrever apenas com línguas no alfabeto latino e não pretende usar fontes diferentes das disponíveis por padrão no  $\text{\LaTeX}$ , qualquer das três versões modernas ( $\text{\pdfLaTeX}$ ,  $\text{\XeLaTeX}$  e  $\text{\LuaLaTeX}$ ) é adequada. Se você pretende usar outros alfabetos ou se gostaria de escolher fontes diferentes, use  $\text{\LuaLaTeX}$ .

## 3.12 Limitações do L<sup>A</sup>T<sub>E</sub>X e Algumas Dicas

Como qualquer ferramenta, L<sup>A</sup>T<sub>E</sub>X tem limitações e características indesejáveis:

- A linguagem é muito prolixa: é bastante tedioso escrever coisas como “\begin{itemize}” etc. Linguagens como asciidoc/asciidoc (asciidoc.org), markdown (daringfireball.net/projects/markdown), bookdown (bookdown.org) e sphinx (sphinx-doc.org) operam de maneira similar a L<sup>A</sup>T<sub>E</sub>X, mas sua sintaxe é bem mais enxuta. Elas funcionam muito bem para a geração de páginas web, mas L<sup>A</sup>T<sub>E</sub>X oferece mais recursos e geralmente produz resultados impressos melhores.
- L<sup>A</sup>T<sub>E</sub>X gera muitas mensagens pouco importantes durante o processamento do documento, o que dificulta a identificação de problemas. Além disso, quando ocorrem erros durante esse processamento, as mensagens explicativas de L<sup>A</sup>T<sub>E</sub>X muitas vezes são confusas ou, pior, não indicam o problema real que causou a falha.
- L<sup>A</sup>T<sub>E</sub>X procura ser uma linguagem *declarativa*, ou seja, os comandos buscam expressar o que se deseja e não como fazer algo (“este texto é um título” e não “pule duas linhas, selecione uma fonte maior, escreva este texto, pule mais duas linhas e selecione a fonte de tamanho padrão”). No entanto, ela é insuficiente em algumas situações, obrigando o usuário a utilizar vários comandos, às vezes obscuros, para obter resultados relativamente simples.
- Há diversas packages para personalizar os aspectos básicos da formatação final do documento, como o tipo de fonte, tamanho dos títulos das seções, espaçamento etc. No entanto, quando se quer fazer modificações maiores, é preciso lidar com partes complexas da linguagem e diversos comportamentos surpreendentes.
- Às vezes há incompatibilidades entre packages; em alguns casos, isso pode ser contornado mudando a ordem em que elas são carregadas, mas em outros pode simplesmente não ser possível combiná-las.
- A colocação automática dos *floats* em geral funciona bem, mas nem sempre. Isso acontece porque L<sup>A</sup>T<sub>E</sub>X decide o posicionamento de cada *float* individualmente, sem levar em conta os próximos *floats*, e nunca reavalia essa decisão. No exemplo da Seção 3.5, se a ordem “Figura 5, Tabela 3, Figura 6” for aceitável, esse vai ser o resultado, mesmo que a ordem “Tabela 3, Figura 5, Figura 6” seja melhor. Apenas se não for possível encontrar um lugar aceitável para a Figura 5 imediatamente (ou seja, na página atual) é que L<sup>A</sup>T<sub>E</sub>X processa os *floats* seguintes e, depois, procura novamente um lugar para ela. Por isso, depois que seu trabalho estiver finalizado, vale a pena avaliar se a colocação dos *floats* pode ser melhorada; se sim, mudar o lugar em que eles são definidos no documento (veja algumas dicas em MITTELBAUGH, 2014) pode fazer L<sup>A</sup>T<sub>E</sub>X gerar um resultado melhor (mas lembre-se que isso só faz sentido depois que o documento estiver pronto, pois qualquer mudança no texto pode mudar totalmente a posição final dos *floats*).
- O algoritmo que L<sup>A</sup>T<sub>E</sub>X usa para quebrar páginas é excelente, minimizando linhas órfãs ou viúvas e garantindo uma distribuição homogênea do texto na página. No entanto, ele não utiliza um recurso comumente usado por editores profissionais, que

é mudar o tamanho de algumas páginas para melhorar a distribuição geral do texto. Esse é um último recurso, mas que muitas vezes pode ser bastante positivo. Ainda assim, se houver quebras de página ruins no seu texto final, você pode usar essa estratégia manualmente. Ao invés de comandos como `\pagebreak` ou `\newpage`, o mais adequado é usar `\enlargethispage{\baselineskip}`. Esse comando instrui  $\text{\LaTeX}$  a fazer a página ligeiramente maior, tornando possível acomodar mais uma linha (`-1\baselineskip` faz a página ficar com uma linha a menos). Em documentos frente e verso, lembre-se de sempre garantir que a página adjacente também tenha seu tamanho modificado para que a alteração não seja tão perceptível. Um outro truque às vezes útil é aplicar o comando `\looseness=1` (ou `-1`) a um parágrafo, que faz  $\text{\LaTeX}$  tentar reorganizar as quebras de linha de maneira a fazer o parágrafo ter uma linha a mais (ou a menos), se isso for possível.

- Como muitos outros sistemas de texto,  $\text{\LaTeX}$  pode usar mais de um padrão para a codificação de caracteres acentuados (através da configuração da package `inputenc`). Alguns anos atrás, o mais comum era o ISO-8859-1, também conhecido como `latin1` (esse é o nome usado no  $\text{\LaTeX}$ ) ou `Windows-1252`; atualmente, o mais comum é o UTF-8. No entanto, usuários que escrevem apenas em língua inglesa às vezes não configuram seus sistemas para usar qualquer tipo de caracter acentuado. De maneira geral, é simples reconhecer e resolver os problemas causados por inconsistências na codificação (seja trocando a opção de `inputenc`, seja recodificando o arquivo), mas arquivos “.bib” são um caso especial: `biblatex` (usado neste modelo) funciona normalmente com caracteres acentuados, mas `bibtex` oficialmente não tem suporte a eles (embora em geral funcione corretamente). Além disso, é bastante comum que arquivos desse tipo sejam compartilhados por várias pessoas, com diferentes configurações. Para evitar problemas com os acentos nesse caso, uma possibilidade é representar os caracteres acentuados usando comandos  $\text{\LaTeX}$ : `\{a}` para á, `\{c}` para cedilha etc., independentemente da codificação usada no texto<sup>10</sup>.
- As classes padrão (`book`, `article` etc.) não foram criadas para serem facilmente modificadas, o que deu origem a inúmeras packages voltadas para possibilitar a personalização de diversos aspectos da apresentação final do documento. Esse mecanismo não é ideal, por diversas razões. Por conta disso, existe um conjunto de versões alternativas dessas classes (`scrbook` no lugar de `book`, `scrartcl` no lugar de `article` etc.) chamado KOMA-Script, com mais recursos e mais possibilidades de customização. A classe `memoir` tem o mesmo objetivo, mas procura dar suporte a livros e artigos com uma única classe. Ambas abordagens são muito boas, mas a maioria dos modelos usados por revistas e outras publicações é baseada nas classes padrão. A versão 3 de  $\text{\LaTeX}$  está em desenvolvimento com vistas a resolver boa parte dos problemas atuais do sistema, mas ainda deve demorar muitos anos para ficar pronta. `ConTeXt` é um “irmão mais novo” de  $\text{\LaTeX}$  com diversas vantagens, mas com sintaxe diferente e que ainda não é tão popular.

<sup>10</sup>Você pode consultar os comandos desse tipo mais comuns em [en.wikibooks.org/wiki/LaTeX/Special\\_Characters](http://en.wikibooks.org/wiki/LaTeX/Special_Characters). Observe que a dica sobre o pingo do *i* não é mais válida atualmente; basta usar `\{i}`.

# Chapter 4

## Alguns Exemplos de Comandos $\text{\LaTeX}$

### 4.1 Bibliografia e Referências

A documentação do pacote biblatex (LEHMAN *et al.*, 2018) é bastante extensa e explica (nas Seções 2.1.1 e 2.2.2) os diversos tipos de documento suportados, bem como o significado de cada campo. Na prática, às vezes é preciso fazer escolhas sobre o que incluir na descrição de um item bibliográfico e muitas vezes é mais fácil aprender copiando exemplos já existentes, como estes (consulte o arquivo `bibliografia.bib` para ver como foi criado o banco de dados e a bibliografia na página 35 para ver o resultado impresso):

- @Book: KNUTH *et al.*, 1996.
- @Article (em periódico): MITTELBACH, 2014.
- @InProceedings (ou @Conference): ALVES *et al.*, 2003.
- @InCollection (capítulo de livro ou coletânea): BABAÖGLU and MARZULLO, 1993.
- @PhdThesis: GARCIA, 2001.
- @MastersThesis: SCHMIDT, 2003.
- @Techreport: ALVISI *et al.*, 1999.
- @Manual: LEHMAN *et al.*, 2018.
- @Misc: ALLCOCK, 2003.
- @Online (para referência a artigo *online*): FOWLER, 2004.
- @Online (para referência a página web): FSF, 2007.

### 4.2 Modo Matemático

O modo matemático do  $\text{\LaTeX}$  tem sintaxe própria, mas ela não é complicada e há bastante documentação *online* a respeito. Por exemplo, “massa e energia são grandezas relacionadas pela Equação  $E = mc^2$ , definida inicialmente por Einstein”, ou ainda “equações de segundo grau (Equação 4.1) são estudadas no ensino médio. As raízes de uma equação de segundo grau podem ser encontradas por (4.2) — a fórmula de Bháskara. O valor do discriminante  $\Delta$  (Equação 4.3) determina se a equação tem zero, uma ou duas raízes reais



distintas”. Observe que, quando um parágrafo termina com um símbolo, pode ser boa ideia usar um espaço não-separável (com “~”) para evitar que ele fique sozinho na última linha (por exemplo, “O discriminante é denotado por~ $\Delta$ ”).

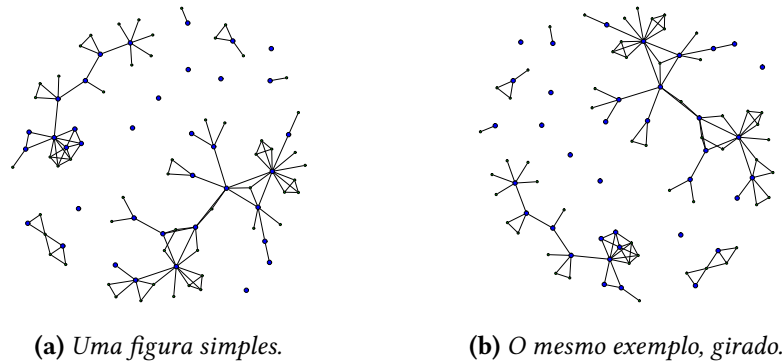
$$ax^2 + bx + c = y \quad \forall x \in \mathbb{R} \quad (4.1)$$

$$y = 0 \Leftrightarrow x = \frac{-b \pm \sqrt{\Delta}}{2a} \Leftrightarrow x \text{ é raiz da equação} \quad (4.2)$$

$$\Delta \text{ (delta)} = b^2 - 4ac \quad (4.3)$$

### 4.3 *Floats* (Tabelas e Figuras)

Evidentemente,  $\text{\LaTeX}$  permite inserir figuras no texto; além disso, ele também permite girá-las e criar subfiguras (com sublegendas), como no exemplo da Figura 4.1, que inclui as subfiguras 4.1a e 4.1b.



**Figure 4.1:** *Exemplo de subfiguras.*

Uma “figura”, na verdade, pode ser qualquer tipo de conteúdo ilustrativo (um exemplo interessante é o cronograma mostrado na Figura 4.2) mas, com a *package* float, também é possível definir ambientes específicos para cada tipo de conteúdo adicional (cada um com numeração independente), como é o caso do Programa 4.1. Há mais informações e dicas sobre recursos específicos para inclusão de código-fonte e pseudocódigo no Apêndice A<sup>1</sup>.

$\text{\LaTeX}$  pode importar gráficos gerados por matplotlib e por gnuplot como qualquer outra imagem, mas nesse caso a fonte usada nesses gráficos provavelmente será diferente do corpo do texto. Conforme mencionado na Seção 3.8, há mecanismos para resolver esse problema<sup>2</sup>, como pode ser visto na Figura 4.3.

<sup>1</sup>Observe que o nome do Apêndice (“A”) foi impresso em uma linha separada, o que não é muito bom visualmente. Para evitar que isso aconteça (não só no final do parágrafo, mas em qualquer quebra de linha), faça o que já foi discutido na Seção 4.2 sobre símbolos matemáticos: utilize um espaço não-separável para fazer referências a figuras, tabelas, seções etc.: “...no Apêndice~\ref{ap:pseudocode}”.

<sup>2</sup>Você pode se interessar também pela package gnuplottex.



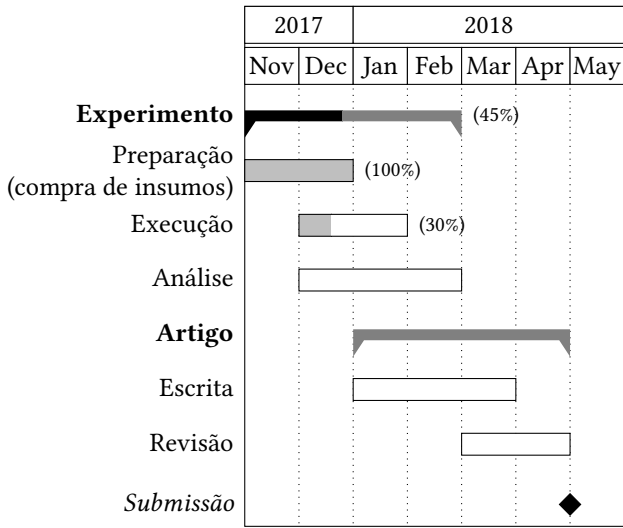


Figure 4.2: Exemplo de cronograma.

**Program 4.1** Exemplo de laço em Java.

```
1  for (i = 0; i < 20; i++)
2  {
3      // Comentário
4      System.out.println("Mensagem...");
5  }
```

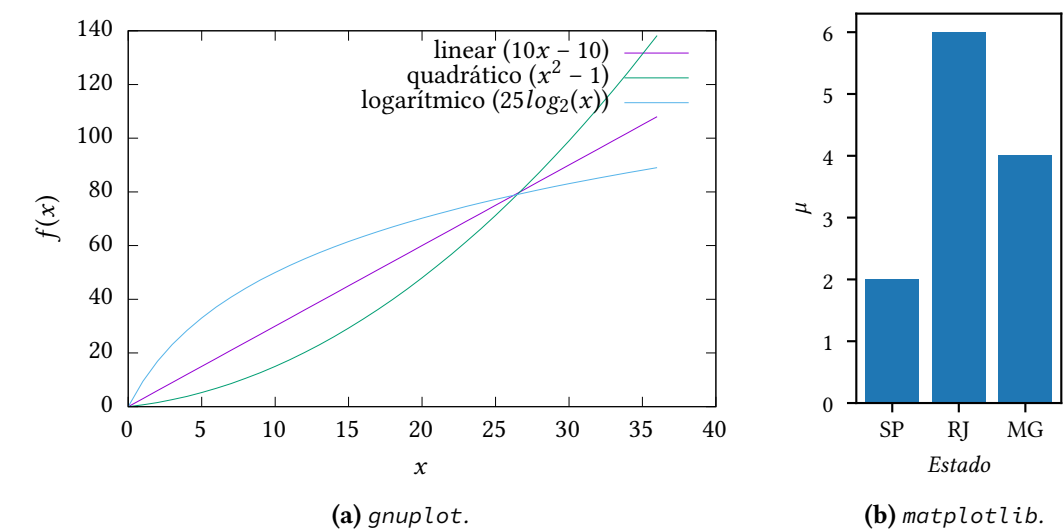


Figure 4.3: Exemplos de gráficos gerados externamente

Finalmente, talvez você precise organizar a apresentação da informação na forma de tabelas<sup>3</sup>; Um exemplo simples é a Tabela 4.1.

Código	Abreviatura	Nome completo
A	Ala	Alanina
C	Cys	Cisteína
...	...	...
W	Trp	Triptofano
Y	Tyr	Tirosina

(a) Com linhas de cores alternadas.

Código	Abreviatura	Nome completo
A	Ala	Alanina
C	Cys	Cisteína
...	...	...
W	Trp	Triptofano
Y	Tyr	Tirosina

(b) Com cabeçalhos girados.

**Table 4.1:** Exemplos de tabelas (códigos, abreviaturas e nomes dos aminoácidos).

Se a tabela tem muitas linhas e, portanto, não cabe em uma única página, é possível fazê-la continuar ao longo de várias páginas com a *package* *longtable*, como é o caso da Tabela 4.2. Nesse caso, a tabela não é um *float* e, portanto, ela aparece de acordo com a sequência normal do texto. Se, além de muito longa, a tabela for também muito larga, você pode usar o comando *landscape* (da *package* *pdflscape*) em conjunto com *longtable* para imprimi-la em modo paisagem ao longo de várias páginas. A Tabela 4.2 tem essa configuração comentada; experimente des-comentar as linhas correspondentes<sup>4</sup>.

Lim.	MGWT			AMI			Spectrum de Fourier			Caract. espectrais		
	Sn	Sp	AC	Sn	Sp	AC	Sn	Sp	AC	Sn	Sp	AC
1	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08
2	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09
3	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
4	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
5	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11
6	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12
7	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.13
8	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13
9	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14
10	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
11	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
12	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16
13	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17
14	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17

continua →

**Table 4.2:** Exemplo de tabela com valores numéricos.

<sup>3</sup>Para defini-las com  $\LaTeX$ , pode valer a pena usar o sítio [www.tablesgenerator.com](http://www.tablesgenerator.com).

<sup>4</sup>Observe que, nesse caso, vai sempre haver uma quebra de página no texto para fazer a tabela começar em uma página em modo paisagem.

Lim.	MGWT			AMI			Spectrum de Fourier			Caract. espectrais		
	Sn	Sp	AC	Sn	Sp	AC	Sn	Sp	AC	Sn	Sp	AC
15	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18
16	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19
17	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19
18	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20
19	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21
20	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
21	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
23	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
24	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
25	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
26	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
27	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
28	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
29	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
30	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
31	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
32	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
33	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
34	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
35	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
36	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
37	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
38	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
39	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22
40	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22

**Table 4.2:** Exemplo de tabela com valores numéricos.

Tabelas mais complexas são um tanto trabalhosas em  $\text{\LaTeX}$ ; a Tabela 4.3 mostra como construir uma tabela em forma de ficha. Além de complexa, ela é larga e, portanto, deve ser impressa em modo paisagem. No entanto, usamos um outro mecanismo para girar a tabela: o comando `sidewaystable` (da *package* `rotating`). Com esse mecanismo, ela continua sendo um *float* (e, portanto, não força quebras de página no meio do texto), mas sempre é impressa em uma página separada.

Resumindo:

- Se uma tabela cabe em uma página, defina-a como um *float*;
- se cabe em uma página mas é muito larga e precisa ser impressa em modo paisagem, use `sidewaystable` (que também é um *float*);
- se não cabe em uma página por ser muito longa, use `longtable`;
- se não cabe em uma página por ser muito longa e precisa ser impressa em modo paisagem por ser muito larga, use `longtable` em conjunto com `landscape`.

Experimento número:	1	Data:					jan 2017
Título:	Medições iniciais						
Tipo de experimento:	Levantamento quantitativo						
Locais	São Paulo	Rio de Janeiro	Porto Alegre	Recife	Manaus	Brasília	Rio Branco
Valores obtidos	0.2	0.3	0.2	0.7	0.5	0.1	0.4

Table 4.3: Exemplo de tabela similar a uma ficha.

# Appendix A

## Código-Fonte e Pseudocódigo

Com a *package listings*, programas podem ser inseridos diretamente no arquivo, como feito no caso do Programa 4.1, ou importados de um arquivo externo com o comando `\lstinputlisting`, como no caso do Programa A.1.

---

### Program A.1 Máximo divisor comum (arquivo importado).

---

```

1  FUNCTION euclid( $a, b$ )  $\triangleright$  The g.c.d. of  $a$  and  $b$ 
2       $r \leftarrow a \bmod b$ 
3      while  $r \neq 0$   $\triangleright$  We have the answer if  $r$  is 0
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
6           $r \leftarrow a \bmod b$ 
7      end
8      return  $b$   $\triangleright$  The g.c.d. is  $b$ 
9  end
```

---

Trechos de código curtos (menores que uma página) podem ou não ser incluídos como *floats*; trechos longos necessariamente incluem quebras de página e, portanto, não podem ser *floats*. Com *floats*, a legenda e as linhas separadoras são colocadas pelo comando `\begin{program}`; sem eles, utilize o ambiente `programruledcaption` (atenção para a colocação do comando `\label{}`, dentro da legenda), como no Programa A.2<sup>1</sup>:

---

### Program A.2 Máximo divisor comum.

---

```

1  FUNCTION euclid( $a, b$ )  $\triangleright$  The g.c.d. of  $a$  and  $b$ 
2       $r \leftarrow a \bmod b$ 
3      while  $r \neq 0$   $\triangleright$  We have the answer if  $r$  is 0
4           $a \leftarrow b$ 
5           $b \leftarrow r$ 
```

*cont*  $\longrightarrow$

---

<sup>1</sup>listings oferece alguns recursos próprios para a definição de *floats* e legendas, mas neste modelo não os utilizamos.

```

→ cont
6       $r \leftarrow a \bmod b$ 
7      end
8      return  $b$  ▷ The g.c.d. is b
9  end

```

---

Além do suporte às várias linguagens incluídas em listings, este modelo traz uma extensão para permitir o uso de pseudocódigo, útil para a descrição de algoritmos em alto nível. Ela oferece diversos recursos:

- Comentários seguem o padrão de C++ (`//` e `/* ... */`), mas o delimitador é impresso como “▷”.
- “:=”, “<”, “<=”, “>=” e “!=” são substituídos pelo símbolo matemático adequado.
- É possível acrescentar palavras-chave além de “if”, “and” etc. com a opção “`morekeywords={pchave1,pchave2}`” (para um trecho de código específico) ou com o comando `\lstset{morekeywords={pchave1,pchave2}}` (como comando de configuração geral).
- É possível usar pequenos trechos de código, como nomes de variáveis, dentro de um parágrafo normal com `\lstinline{blah}`.
- “`$....$`” ativa o modo matemático em qualquer lugar.
- Outros comandos LaTeX funcionam apenas em comentários; fora, a linguagem simula alguns pré-definidos (`\textit{}`, `\texttt{}` etc.).
- O comando `\label` também funciona em comentários; a referência correspondente (`\ref`) indica o número da linha de código. Se quiser usá-lo numa linha sem comentários, use `/// \label{blah}`; “`///`” funciona como `//`, permitindo a inserção de comandos  $\text{\LaTeX}$ , mas não imprime o delimitador (▷).
- Para suspender a formatação automática, use `\noparse{blah}`.
- Para forçar a formatação de um texto como função, identificador, palavra-chave ou comentário, use `\func{blah}`, `\id{blah}`, `\kw{blah}` ou `\comment{blah}`.
- Palavras-chave dentro de comentários não são formatadas automaticamente; se necessário, use `\func{}`, `\id{}` etc. ou comandos  $\text{\LaTeX}$  padrão.
- As palavras “Program”, “Procedure” e “Function” têm formatação especial e fazem a palavra seguinte ser formatada como função. Funções em outros lugares *não* são detectadas automaticamente; use `\func{}`, a opção “`functions={func1,func2}`” ou o comando “`\lstset{functions={func1,func2}}`” para que elas sejam detectadas.
- Além de funções, palavras-chave, strings, comentários e identificadores, há “`specialidentifiers`”. Você pode usá-los com `\specialid{blah}`, com a opção “`specialidentifiers={id1,id2}`” ou com o comando “`\lstset{specialidentifiers={id1,id2}}`”.

# Annex A

## Perguntas Frequentes sobre o Modelo<sup>2</sup>

- **Não consigo decorar tantos comandos!**

Use a colinha que é distribuída juntamente com este modelo ([gitlab.com/ccsl-usp/modelo-latex/raw/master/pre-compilados/colinha.pdf?inline=false](https://gitlab.com/ccsl-usp/modelo-latex/raw/master/pre-compilados/colinha.pdf?inline=false)).

- **Por que tantos arquivos?**

O preâmbulo  $\text{\LaTeX}$  deste modelo é muito longo; as partes que normalmente não precisam ser modificadas foram colocadas no diretório `extras`, juntamente com alguns arquivos acessórios. Já os arquivos de conteúdo (capítulos, anexos etc.) foram divididos de maneira que seja fácil para você atualizar o modelo (copiando os novos arquivos ou com um sistema de controle de versões) sem que alterações no conteúdo de exemplo (este texto que você está lendo) causem conflitos com o seu próprio texto.

- **As figuras e tabelas são colocadas em lugares ruins.**

Veja a discussão a respeito na Seção 3.12.

- **Estou tendo problemas com caracteres acentuados.**

Veja a discussão a respeito na Seção 3.12.

- **Existe algo específico para citações de páginas web?**

Biblatex define o tipo “online”, que deve ser usado para materiais com título, autor etc., como uma postagem ou comentário em um blog, um gráfico ou mesmo uma mensagem de email para uma lista de discussão. Bibtex, por padrão, não tem um tipo específico para isso; com ele, normalmente usa-se o campo “howpublished” para especificar que se trata de um recurso *online*. Se o que você está citando não é algo determinado com título, autor etc. mas sim um sítio (como uma empresa ou um produto), pode ser mais adequado colocar a referência apenas como nota de rodapé e não na lista de referências; nesses casos, algumas pessoas acrescentam uma segunda lista de referências especificamente para recursos *online* (biblatex permite criar múltiplas bibliografias). Já artigos disponíveis *online* mas que fazem

---

<sup>2</sup>Esta seção não é de fato um anexo, mas sim um apêndice; ela foi definida desta forma apenas para servir como exemplo de anexo.

parte de uma publicação de formato tradicional (mesmo que apenas *online*), como os anais de um congresso, devem ser citados por seu tipo verdadeiro e apenas incluir o campo “url” (não é nem necessário usar o comando `\url{}`), aceito por todos os tipos de documento do bibtex/biblatex.

- **Aparece uma folha em branco entre os capítulos.**

Essa característica foi colocada propositalmente, dado que todo capítulo deve (ou deveria) começar em uma página de numeração ímpar (lado direito do documento). Se quiser mudar esse comportamento, acrescente “openany” como opção da classe, i.e., `\documentclass[openany,...]{book}`.

- **É possível resumir o nome das seções/capítulos que aparece no topo das páginas e no sumário?**

Sim, usando a sintaxe `\section[mini-titulo]{titulo enorme}`. Isso é especialmente útil nas legendas (*captions*) das figuras e tabelas, que muitas vezes são demasiadamente longas para a lista de figuras/tabelas.

- **Existe algum programa para gerenciar referências em formato bibtex?**

Sim, há vários. Uma opção bem comum é o JabRef; outra é usar Zotero ou Mendeley e exportar os dados deles no formato .bib.

- **Posso usar pacotes L<sup>A</sup>T<sub>E</sub>X adicionais aos sugeridos?**

Com certeza! Você pode modificar os arquivos o quanto desejar, o modelo serve só como uma ajuda inicial para o seu trabalho.

- **Como faço para usar o Makefile (comando make) no Windows?**

Lembre-se que a ferramenta recomendada para compilação do documento é o latexmk, então você não precisa do make. Mas, se quiser usá-lo, você pode instalar o MSYS2 ([www.msys2.org](http://www.msys2.org)) ou o Windows Subsystem for Linux (procure as versões de Linux disponíveis na Microsoft Store). Se você pretende usar algum dos editores sugeridos, é possível deixar a compilação a cargo deles, também dispensando o make.

- **Como eu faço para...**

Leia os comentários dos arquivos “tese.tex” e outros que compõem este modelo, além do tutorial (Capítulo 3) e dos exemplos do Capítulo 4; é provável que haja uma dica neles ou, pelo menos, a indicação da *package* relacionada ao que você precisa.



## References

- [ALLCOCK 2003] William ALLCOCK. *GridFTP Protocol Specification*. Global Grid Forum Recommendation (GFD.20). 2003 (cit. on p. 25).
- [ALON 2009] Uri ALON. “How To Choose a Good Scientific Problem”. In: *Molecular Cell* 35.6 (Sept. 2009), pp. 726–728. DOI: [10.1016/j.molcel.2009.09.013](https://doi.org/10.1016/j.molcel.2009.09.013) (cit. on p. 5).
- [ALVES *et al.* 2003] Carlos E. R. ALVES, Edson N. CÁCERES, Frank DEHNE, and Siang W. SONG. “A Parallel Wavefront Algorithm for Efficient Biological Sequence Comparison”. In: *ICCSA’03: The 2003 International Conference on Computational Science and its Applications*. Springer-Verlag, May 2003, pp. 249–258 (cit. on p. 25).
- [ALVISI *et al.* 1999] Lorenzo ALVISI, Elmootazbellah ELNOZAHY, Sriram S. RAO, Syed A. HUSAIN, and Asanka Del MEL. *An Analysis of Communication-Induced Checkpointing*. Tech. rep. TR-99-01. Austin, USA: Department of Computer Science, University of Texas at Austin, 1999 (cit. on p. 25).
- [BABAOGU and MARZULLO 1993] Ozalp BABAOGU and Keith MARZULLO. “Consistent Global States of Distributed Systems: Fundamental Concepts and Mechanisms”. In: *Distributed Systems*. Ed. by Sape MULLENDER. 2nd ed. 1993, pp. 55–96 (cit. on p. 25).
- [BOOTH *et al.* 2008] Wayne C. BOOTH, Gregory G. COLOMB, and Joseph M. WILLIAMS. *The Craft of Research*. The University of Chicago Press, 2008 (cit. on p. 5).
- [CARLIS 2009] John V. CARLIS. *Design: The Key to Writing (and Advising) a One-Draft Ph.D Dissertation*. 2009. URL: [www-users.cs.umn.edu/~carlis/one-draft.pdf](http://www-users.cs.umn.edu/~carlis/one-draft.pdf) (visited on 11/10/2017) (cit. on p. 4).
- [DALY 2010] Patrick W. DALY. *Reference sheet for natbib usage*. Sept. 13, 2010. URL: [mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf](http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf) (visited on 12/20/2018) (cit. on pp. 5, 18).
- [ECO 2009] Umberto ECO. *Como se Faz uma Tese*. 22nd ed. Tradução Gilson Cesar Cardoso de Souza. Perspectiva, 2009 (cit. on p. 4).
- [FOWLER 2004] Martin FOWLER. *Is Design Dead?* May 2004. URL: [martinfowler.com/articles/designDead.html](http://martinfowler.com/articles/designDead.html) (visited on 01/30/2010) (cit. on p. 25).

- [FSF 2007] FREE SOFTWARE FOUNDATION. *GNU General Public License*. 2007. URL: [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html) (visited on 01/30/2010) (cit. on p. 25).
- [GARCIA 2001] Islene C. GARCIA. “Visões Progressivas de Computações Distribuídas”. PhD thesis. Campinas, Brasil: Instituto de Computação, Universidade de Campinas, Dec. 2001 (cit. on p. 25).
- [HIGHAM 1998] Nicholas J. HIGHAM. *Handbook of Writing for the Mathematical Sciences*. 2nd ed. SIAM: Society for Industrial and Applied Mathematics, Aug. 1998 (cit. on p. 5).
- [KNUTH *et al.* 1996] Donald E. KNUTH, Tracy LARRABEE, and Paul M. ROBERTS. *Mathematical Writing*. The Mathematical Association of America, Sept. 1996 (cit. on pp. 5, 25).
- [LEHMAN *et al.* 2018] Philipp LEHMAN, Philip KIME, Moritz WEMHEUER, Audrey BORUVKA, and Joseph WRIGHT. *The bibl<sub>at</sub>ex Package*. Oct. 30, 2018. URL: [mirrors.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf](http://mirrors.ctan.org/macros/latex/contrib/biblatex/doc/biblatex.pdf) (visited on 12/20/2018) (cit. on pp. 19, 25).
- [MITTELBAACH 2014] Frank MITTELBAACH. “How to influence the position of float environments like figure and table in  $\text{\TeX}$ ?” In: *TUGboat. Communications of the  $\text{\TeX}$  Users Group* 35.3 (2014). URL: [tug.org/TUGboat/tb35-3/tb111mitt-float.pdf](http://tug.org/TUGboat/tb35-3/tb111mitt-float.pdf) (visited on 01/09/2020) (cit. on pp. 23, 25).
- [ROBERTS 2010] Carol M. ROBERTS. *The Dissertation Journey*. 2nd ed. Thousand Oaks, CA: Corwin, 2010 (cit. on p. 5).
- [SCHMIDT 2003] Rodrigo M. SCHMIDT. “Coleta de Lixo para Protocolos de *Checkpointing*”. MA thesis. Campinas, Brasil: Instituto de Computação, Universidade de Campinas, Oct. 2003 (cit. on p. 25).
- [TUFTE 2001] Edward TUFTE. *The Visual Display of Quantitative Information*. 2nd ed. Graphics Press, May 2001 (cit. on p. 5).
- [SIBiUSP 2009] UNIVERSIDADE DE SÃO PAULO — SISTEMA INTEGRADO DE BIBLIOTECAS. *Diretrizes para Apresentação de Dissertações e Teses da USP: Documento Eletrônico e Impresso*. 2009. URL: [www.teses.usp.br/index.php?option=com\\_content&view=article&id=52&Itemid=67](http://www.teses.usp.br/index.php?option=com_content&view=article&id=52&Itemid=67) (visited on 11/10/2017) (cit. on p. 5).
- [WAZLAWICK 2009] Raul S. WAZLAWICK. *Metodologia de Pesquisa em Ciência da Computação*. 1st ed. Campus, 2009 (cit. on p. 5).
- [ZOBEL 2004] Justin ZOBEL. *Writing for Computer Science: The Art of Effective Communication*. 2nd ed. Springer, 2004 (cit. on p. 5).