

UNIVERSIDADE DE SÃO PAULO  
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

**Simulação numérica do problema de  
N-corpos gravitacional**

***Projeto de trabalho de formatura***

Octavio Augusto Potalej

MONOGRAFIA FINAL

MAP 2080 — TRABALHO DE  
FORMATURA

Supervisor: Profº. Drº. Eduardo Colli

São Paulo  
2024

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0  
(Creative Commons Attribution 4.0 International License)*

## Resumo

Octavio Augusto Potálej. **Simulação numérica do problema de N-corpos gravitacional: Projeto de trabalho de formatura.** Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2024.

[illegible]

**Palavras-chave:** Problema de N-corpos. Simulação numérica. Integração numérica.



# Abstract

Octavio Augusto Potalej. **Numerical simulation of gravitational N-body problem: Graduation project.** Capstone Project Report (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2024.

[illegible]

**Keywords:** N-body problem. Numerical simulation. Numerical integration.



# Capítulo 1

## Introdução

**A introdução precisa ter um CONTEXTO, um OBJETIVO e uma HIPÓTESE.**

Desde o início da formalização da Mecânica por Isaac Newton, diversos sistemas mecânicos receberam maior ou menor atenção a depender de sua relevância. Um dos mais famosos é o chamado *Problema de N-corpos* (PNC). O problema de N-corpos consiste na suposição de  $N$  partículas dispostas no vácuo e interagindo a distância através da força gravitacional ou da eletromagnética. A dificuldade - e a denominação *problema* - aparece quando  $N$  é um número suficientemente grande, pois resolver as equações de movimento analiticamente passa a ser, no geral, impossível. De fato, o sistema passa a ser um problema já em  $N = 3$ .

Alguns resultados qualitativos a respeito de PNCs com valores iniciais específicos foram obtidos e descritos nas obras de Lagrange, Painlevé e Sundman, em especial acerca de colisões entre partículas e a existência de soluções. Além disso, algumas teorias de gravidade baseadas em Relatividade Geral, como a Dinâmica de Formas (*Shape Dynamics*), possuem simplificações descritas pelo PNC e que trazem resultados gerais interessantes, como o modelo de Barbour-Bertotti (BB) e sua redução dimensional do Problema.

Na aplicação geral do problema, como no lançamento de foguetes e sondas e no estudo de sistemas planetários simples, o uso de Análise Numérica para a integração das equações de movimento do PNC é fundamental. A infinitude de métodos existentes faz ser necessário o estudo de quais tipos de métodos numéricos se adequam melhor ao objetivo da simulação, que nem sempre vão no caminho da precisão.

Simulações de grande escala, como o AbacusSummit, visam mais o panorama geral do que o local, ao contrário de simulações de rotas de sondas que passem por estilingues gravitacionais, como as Voyagers. Assim, enquanto as simulações do primeiro tipo podem se utilizar de métodos numéricos com menor precisão mas maior rapidez, as do segundo tipo necessitam de métodos mais precisos e consequentemente mais custosos.

De forma geral, há uma categoria de métodos voltados para a integração numérica de sistemas mecânicos fechados, chamados *Métodos Simpléticos* (MS). Tais métodos baseiam-se na estrutura simplética de sistemas mecânicos, como o PNC, e utilizam da conservação

de alguns valores, como a energia total, para fornecer uma maior estabilidade da solução e a garantia de verossimilhança para simulações com um longo período de tempo. Isso é particularmente interessante em simulações planetárias, nas quais as órbitas são, no geral, estáveis e quase periódicas.

Para todo tipo de propósito, no entanto, é um desafio encontrar os métodos e ferramentas mais adequados, pois para cada método existe uma variedade de programas e linguagens de programação que se adequam em maior ou menor medida.

O objetivo deste trabalho, assim, é estudar a simulação numérica do PNC e a possibilidade de utilizá-la na busca de resultados qualitativos do Problema. Estudaremos a formalização e alguns resultados interessantes já descobertos do PNC, classes de métodos numéricos e sua adequação ao Problema, correções numéricas a posteriori e a possibilidade de estender os resultados obtidos para PNCs mais elaborados, como com colisões elásticas.

Ao final, pretendemos mostrar que os métodos simpléticos são os mais adequados para simulações de propósito geral, e que a extensão dos resultados é possível se a complexificação mantiver a estrutura simplética do PNC.



## Capítulo 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:
  - Capítulos, apêndices etc. no diretório `conteudo`
  - `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/imeusp-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)
  - `extras/bibconfig.tex` (configuração da bibliografia)
- Outros arquivos auxiliares (geralmente não precisam ser editados):
  - `extras/languages.tex` (suporte a múltiplas línguas)
  - `extras/imeusp-thesis.tex` (formatação da capa e páginas preliminares)

---

<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/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)
- `froufrou.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)
- `extras/texlogsieve` e seu arquivo de configuração `extras/texlogsieverc` (mostra as mensagens mais importantes geradas durante a compilação)



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 uma coleção com todos eles juntos. Há duas coleções desse tipo disponíveis: MiK<sub>T</sub><sub>E</sub>X ([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><sub>E</sub>X ([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><sub>E</sub>X.

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

---

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

instalação completa do  $\LaTeX$  seja relativamente grande (perto de 5GB), em geral vale a pena instalar a maior parte dos componentes. Se você preferir uma instalação mais “enxuta”, não deixe de incluir tudo que é necessário 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 MiKTeX ou T<sub>E</sub>XLive. 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 $\LaTeX$

Há muito material sobre  $\LaTeX$  na Internet, mas também há muita informação obsoleta (incluindo trechos da própria documentação oficial!). 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 3.12) 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.

Como dito anteriormente,  $\LaTeX$  é, na verdade, um conjunto de programas e, em geral, instalamos coleções pré-prontas com todos eles. Essas coleções (T<sub>E</sub>XLive e MiKTeX) contêm também a documentação das *packages* incluídas: Basta digitar texdoc nome-da-package (T<sub>E</sub>XLive) ou mthelp nome-da-package (MiKTeX) para ter acesso à documentação correspondente. texdoc/mthelp incluem também alguns tutoriais e textos introdutórios.

Um possível caminho para o aprendizado é começar com o Capítulo 3 deste modelo e o conteúdo em [overleaf.com/learn](http://overleaf.com/learn), que tem escopo similar mas também inclui várias páginas sobre como utilizar recursos específicos. Após esse contato inicial, o tutorial em [tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf](http://tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf) é bastante abrangente e detalhado. Depois que você estiver razoavelmente familiarizado com a linguagem, utilize o manual de referência que pode ser acessado em [latexref.xyz](http://latexref.xyz) ou com texdoc latex2e (disponível também em francês, com texdoc latex2e-fr.pdf, e em espanhol, com texdoc latex2e-es.pdf). Para os principais comandos do modo matemático, veja texdoc undergradmath.

Existem também 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*

<sup>3</sup> Uma versão não-final está disponível em [www2.mps.mpg.de/homes/daly/GTL/gtl\\_20030512.pdf](http://www2.mps.mpg.de/homes/daly/GTL/gtl_20030512.pdf).

importantes de L<sup>A</sup>T<sub>E</sub>X, servindo tanto para o aprendizado quanto como material de referência.

Para dúvidas pontuais, o sítio [tex.stackexchange.com](http://tex.stackexchange.com) é um fórum de perguntas e respostas sobre L<sup>A</sup>T<sub>E</sub>X muito útil, pois os principais desenvolvedores do sistema participam das discussões, e o sítio [texfaq.org](http://texfaq.org) é bastante abrangente e atualizado.

### 2.2.1 Outros Recursos

Existem inúmeras alternativas aos materiais citados acima; outros exemplos de textos introdutórios são [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 canal [youtube.com/c/anteroneves](https://youtube.com/c/anteroneves) tem vários vídeos instrutivos em português. `texdoc/mthelp` incluem ainda opções 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) 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 [ctan.org](http://ctan.org) é o repositório semi-oficial das *packages* L<sup>A</sup>T<sub>E</sub>X e sua documentação; T<sub>E</sub>XLive e MiK<sub>T</sub><sub>E</sub>X são construídas a partir do que está nesse site, então a última versão estável de qualquer *package* (e da documentação acessível com `texdoc/mthelp`) em geral está ali.

A documentação de referência mais importante sobre os recursos matemáticos é acessível com `texdoc amsmath`, `texdoc amsthm` e `texdoc mathtools`; `texdoc maths-symbols` agrega os símbolos matemáticos disponíveis. Para uma lista completa de todos os símbolos disponíveis com L<sup>A</sup>T<sub>E</sub>X, use `texdoc symbols-a4` (esse documento tem mais de 300 páginas!).

`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).

Minúcias sobre o funcionamento interno do sistema 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.

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`).

## Capítulo 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 L<sup>A</sup>T<sub>E</sub>X, Scribus, Adobe InDesign ou outras; processadores de texto convencionais não oferecem o mesmo nível de qualidade dessas ferramentas<sup>1</sup>.

### 3.1 Visão Geral

Com L<sup>A</sup>T<sub>E</sub>X, 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 "latexmk" ("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 L<sup>A</sup>T<sub>E</sub>X).

Em geral, é recomendável usar editores projetados especificamente para trabalhar com L<sup>A</sup>T<sub>E</sub>X; eles utilizam cores para distinguir o texto dos comandos de formatação, automatizam o processo de compilação do documento e oferecem outras comodidades. O mais usado atualmente é o T<sub>E</sub>Xstudio, que é software livre e funciona em Windows, MacOS e Linux. O editor Visual Studio Code ([code.visualstudio.com](https://code.visualstudio.com)) é voltado para programadores e tem uma interface às vezes peculiar para outros usuários, mas em conjunto com a *package* LaTeX Workshop (do editor, não do L<sup>A</sup>T<sub>E</sub>X), é uma boa opção. O mesmo vale para o editor emacs ([www.gnu.org/software/emacs](https://www.gnu.org/software/emacs)) e sua *package* AUCT<sub>E</sub>X. Ainda outra possibilidade são os editores *online*; dentre eles, o overleaf ([www.overleaf.com](https://www.overleaf.com)) é o mais usado.

Um documento L<sup>A</sup>T<sub>E</sub>X 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.

---

<sup>1</sup> A maior diferença (mas não a única) é o algoritmo que divide cada parágrafo em uma série de linhas: T<sub>E</sub>X (desde 1982) e Adobe InDesign (desde 1999) analisam cada parágrafo como um todo, ao invés de uma linha por vez, para obter espaçamentos mais homogêneos e menos palavras hifenizadas.

## 3.2 Estrutura de um Documento $\text{\LaTeX}$

O preâmbulo  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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  $\text{\LaTeX}$  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.

$\text{\LaTeX}$  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 “`%`” no texto é preciso digitar “`\%`”.

Como mencionado anteriormente,  $\text{\LaTeX}$  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  $\text{\LaTeX}$ , 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  $\text{\LaTeX}$ : procure definir a estrutura, não a aparência.

Um exemplo de documento  $\text{\LaTeX}$  simples (lembre-se, “`%`” indica um comentário):

```
% O documento começa com o preâmbulo
% Vamos usar a classe "book" com fonte no tamanho 11pt
\documentclass[11pt]{book}
% Vamos escrever em português do Brasil
\usepackage[brazilian]{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 um bloco ou página de título com os dados 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 3.12), 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>2</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>3</sup>. Assim, embora seja possível gerar o PDF executando apenas `pdflatex nome-do-arquivo.tex`, acostume-se a compilar o documento sempre com `latexmk -pdf 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>4</sup>:

```

\begin{itemize}
  \item Pedra
  \item Papel
  \item Tesoura
\end{itemize}

```

<sup>2</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>3</sup> É possível personalizar o comportamento de `latexmk` com o arquivo de configuração `latexmkrc`.

<sup>4</sup> Observe o uso de espaços no início das linhas com `\item` para deixar a estrutura visualmente mais clara durante a edição.





## 3.6 Referências Cruzadas

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. Dentro do *float*, além da figura em si, em geral é uma boa ideia acrescentar uma legenda com `\caption`. Além disso, é possível inserir um `\label` dentro da legenda para que se possa fazer referência à figura/tabela no texto (com os comandos `\ref` e `\pageref`)<sup>6</sup>.

## 3.7 Referências Bibliográficas e Bibliografia

A geração de bibliografias no L<sup>A</sup>T<sub>E</sub>X é feita através da package `biblatex` e do programa auxiliar `biber`<sup>7</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>8</sup>.
2. A inserção de referências às obras ao longo do texto, usando diferentes comandos dependendo do caso: `\cite`, `\citet`, `\citep` etc. Como já mencionado, esses comandos estão descritos na documentação da package `natbib` (DALY, 2010).
3. A escolha do estilo bibliográfico (usando as opções da package `biblatex`) que formata as citações ao longo do texto e gera a bibliografia automaticamente através do comando `\printbibliography`. 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 `\nocite`.

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, sites 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}`:

<sup>6</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.

<sup>7</sup> Antigamente, usava-se a package `natbib` e o comando auxiliar `bibtex`. O funcionamento geral dos dois mecanismos é similar e o formato do banco de dados de ambos é o mesmo.

<sup>8</sup> É comum criar bancos de dados desse tipo separados por assunto, mas isso não é necessário.

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

Em alguns casos,  $\text{\LaTeX}$  troca as letras maiúsculas definidas em `\title` para minúsculas. Para evitar que isso afete siglas ou nomes próprios, basta colocá-los entre chaves (“Automated Application-Level Checkpointing of {MPI} Programs”).

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 certa que `biblatex` oferece uma solução.

## 3.8 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 `\$`).
- Fórmulas mais longas ou que devem aparecer em um parágrafo separado são inseridas com `\[fórmula\]` (ou `\begin{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 `\quad` ou `\enspace`. Para inserir texto “normal” em uma fórmula matemática, use `\text{texto}` (para texto de fato) ou `\mathit{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 `\begin{equation}`, a fórmula recebe um número (que aparece à direita) ao qual você pode se referir no texto usando os comandos “`\ref`” e “`\eqref`” (“conforme vimos na equação `\ref{eq:bhaskara}`...” ou “de acordo com `\eqref{eq:bhaskara}`...”<sup>9</sup>). `\begin{equation}` (incluindo o `*`) elimina o número e é, portanto, equivalente a `\begin{displaymath}`. Há outros comandos similares, como `align`, `multline` e `gather`, definidos e documentados na package `amsmath`, e todos têm a variante com “`*`”.

### 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 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>, `\babelhyphenation` (permite “ensinar” L<sup>A</sup>T<sub>E</sub>X como hifenizar uma lista de palavras, veja `texdoc babel`; note que, em geral, a hifenização automática de L<sup>A</sup>T<sub>E</sub>X é excelente), `\-` (sugere uma possível hifenização localizada), `\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 L<sup>A</sup>T<sub>E</sub>X 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 L<sup>A</sup>T<sub>E</sub>X (`\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 L<sup>A</sup>T<sub>E</sub>X deste capítulo para alguns exemplos simples.

### 3.10 Formatos de Imagem

Podemos classificar imagens em quatro categorias:

<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`.

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  $\text{\LaTeX}$ , é possível importar imagens fotográficas nos formatos PNG e JPEG e imagens dos demais tipos no formato PDF. Além disso,  $\text{\LaTeX}$  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>10</sup>, e matplotlib, com o *backend* `pgf`<sup>11</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  $\text{\LaTeX}$ .

## 3.11 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`”

<sup>10</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>11</sup> [matplotlib.org/users/pgf.html](http://matplotlib.org/users/pgf.html)

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.

- 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, L<sup>A</sup>T<sub>E</sub>X 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.

### 3.12 Versões do L<sup>A</sup>T<sub>E</sub>X

Assim como há packages para o L<sup>A</sup>T<sub>E</sub>X, o próprio L<sup>A</sup>T<sub>E</sub>X é, na verdade, um conjunto de extensões para o programa T<sub>E</sub>X. Assim, se você encontrar referências a “T<sub>E</sub>X” ou a “plain T<sub>E</sub>X”, basta saber que esse é o sistema que funciona “por baixo” do L<sup>A</sup>T<sub>E</sub>X.

L<sup>A</sup>T<sub>E</sub>X é um sistema em evolução (desde os anos 80!). Uma das consequências disso é que há, na verdade, quatro versões diferentes dele:

1. L<sup>A</sup>T<sub>E</sub>X “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 L<sup>A</sup>T<sub>E</sub>X. Hoje em dia não há boas razões para usar essa versão.
2. pdfL<sup>A</sup>T<sub>E</sub>X, 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 L<sup>A</sup>T<sub>E</sub>X.
3. X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X que, além dos recursos do pdfL<sup>A</sup>T<sub>E</sub>X, opera internamente em UTF-8 (ou seja, funciona melhor com múltiplas línguas) e pode funcionar não só com as fontes adaptadas para o L<sup>A</sup>T<sub>E</sub>X como também com as fontes instaladas no sistema. X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X foi muito importante ao ser lançado, mas atualmente a comunidade está mais empenhada em evoluir o sistema com LuaL<sup>A</sup>T<sub>E</sub>X.
4. LuaL<sup>A</sup>T<sub>E</sub>X, que oferece os mesmos recursos que o X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X e também pode ser estendido internamente com mais facilidade (através da linguagem de programação Lua).

Todas essas versões são instaladas quando você instala L<sup>A</sup>T<sub>E</sub>X na sua máquina. Em geral, se você pretende escrever apenas com línguas no alfabeto latino e não pretende usar fontes diferentes das disponíveis por padrão no L<sup>A</sup>T<sub>E</sub>X, qualquer das três versões modernas (pdfL<sup>A</sup>T<sub>E</sub>X, X<sub>Y</sub>L<sup>A</sup>T<sub>E</sub>X e LuaL<sup>A</sup>T<sub>E</sub>X) é adequada, mas pdfL<sup>A</sup>T<sub>E</sub>X é um pouco mais rápido. Se você pretende usar outros alfabetos, gostaria de escolher fontes diferentes ou precisa de recursos tipográficos avançados (texdoc fontspec, texdoc unicode-math), use LuaL<sup>A</sup>T<sub>E</sub>X.



### 3.13 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.org](http://asciidoc.org)), markdown ([commonmark.org](http://commonmark.org)), bookdown ([bookdown.org](http://bookdown.org)) e reStructuredText ([sphinx-doc.org](http://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 MITTELBACH, 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 funciona bem, minimizando linhas órfãs ou viúvas e garantindo uma distribuição homogênea do texto na página, mas não é excelente. Assim, se houver quebras de página ruins no seu texto final, pode ser útil modificá-las manualmente. Uma técnica usada por editores profissionais é mudar

ligeiramente a altura do texto impresso em algumas páginas, melhorando a distribuição geral do texto. Para isso, ao invés de comandos como `\pagebreak` ou `\newpage`, o mais adequado é usar `\enlargethispage{\baselineskip}` (ou `-1\baselineskip`). Esse comando instrui L<sup>A</sup>T<sub>E</sub>X a fazer a página ligeiramente maior (ou menor), tornando possível acomodar mais uma linha de texto (ou 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 L<sup>A</sup>T<sub>E</sub>X tentar reorganizar as quebras de linha de maneira a fazer o parágrafo ter uma linha a mais (ou a menos), se isso for possível.

- 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 L<sup>A</sup>T<sub>E</sub>X está em desenvolvimento com vistas a resolver boa parte dos problemas atuais do sistema, mas ainda deve demorar muitos anos para ficar pronta. ConT<sub>E</sub>Xt é um “irmão mais novo” de L<sup>A</sup>T<sub>E</sub>X com diversas vantagens, mas com sintaxe diferente e que ainda não é tão popular.



## Capítulo 4

# Alguns Exemplos de Comandos L<sup>A</sup>T<sub>E</sub>X

### 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 29 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): BABAOGLU e 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 L<sup>A</sup>T<sub>E</sub>X 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~ $\Delta$ ”).

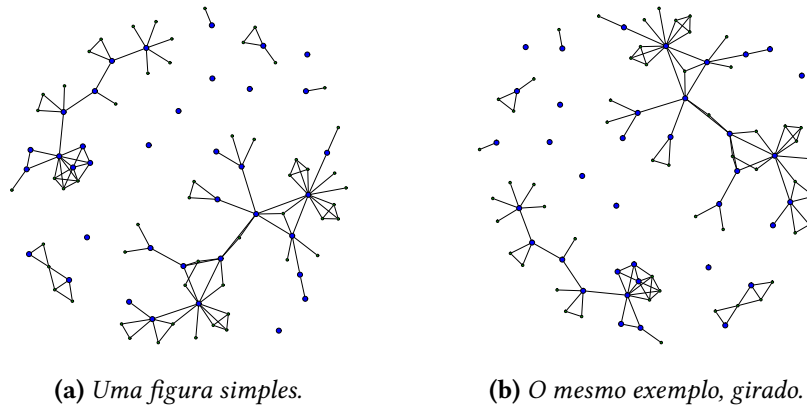
$$ax^2 + bx + c = y \quad \forall x \in \mathbb{R} \quad (4.1)$$

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

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

### 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.



**Figura 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>.

$\text{\LaTeX}$  é capaz de gerar gráficos de excelente qualidade com a *package* pgfplots (há um exemplo comentado neste arquivo; experimente des-comentar para ver o resultado). Também é possível importar gráficos gerados por matplotlib e por gnuplot como qualquer

<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}”.

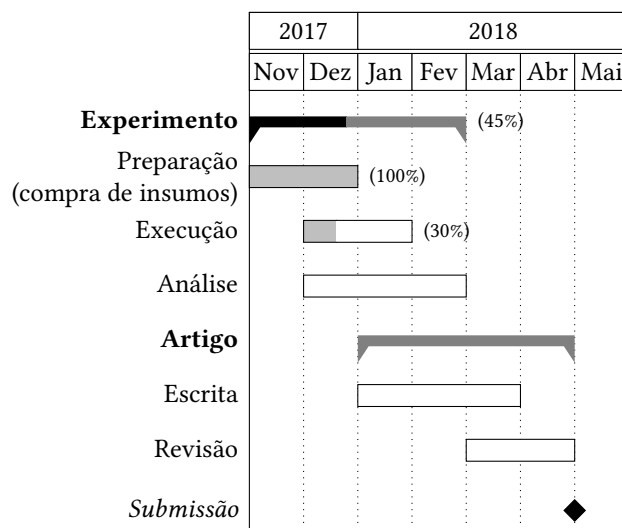


Figura 4.2: Exemplo de cronograma.

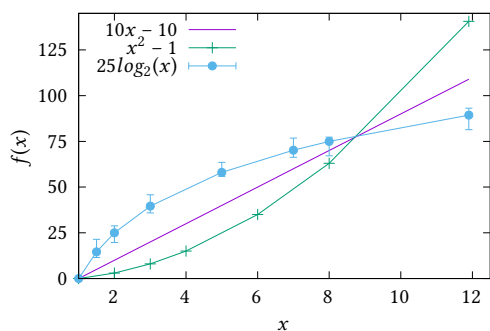
**Programa 4.1** Exemplo de laço em Java.

```

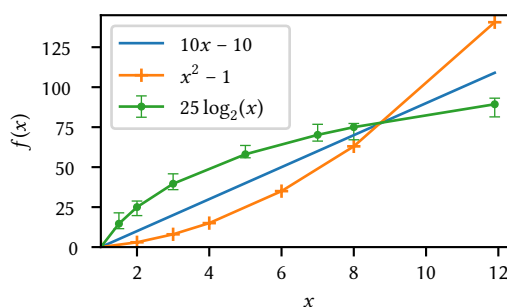
1  for (i = 0; i < 20; i++)
2  {
3      // Comentário
4      System.out.println("Mensagem...");
5  }

```

outra imagem, mas nesse caso a fonte usada nesses gráficos provavelmente será diferente do corpo do texto. Conforme mencionado na Seção 3.10, há mecanismos para resolver esse problema<sup>2</sup>: o resultado pode ser visto na Figura 4.3.



(a) gnuplot.



(b) matplotlib.

Figura 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. Para um resultado visual excelente, não deixe de ler a documentação da *package* booktabs.

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

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

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

(a) Com linhas de cores alternadas.

(b) Com cabeçalhos girados.

Tabela 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>. Ela também demonstra o uso da *package* *siunitx* para alinhar as colunas numéricas pelo separador decimal.

Ângulo		Função					
graus	rads	sen	cos	tan	cotan	sec	cosec
0	0	0	1	0	-	1	-
3	0,0524	0,0523	0,9986	0,0524	19,0811	1,0014	19,1073
6	0,1047	0,1045	0,9945	0,1051	9,5144	1,0055	9,5668
9	0,1571	0,1564	0,9877	0,1584	6,3138	1,0125	6,3925
12	0,2094	0,2079	0,9781	0,2126	4,7046	1,0223	4,8097
15	0,2618	0,2588	0,9659	0,2679	3,7321	1,0353	3,8637
18	0,3142	0,3090	0,9511	0,3249	3,0777	1,0515	3,2361
21	0,3665	0,3584	0,9336	0,3839	2,6051	1,0711	2,7904
24	0,4189	0,4067	0,9135	0,4452	2,2460	1,0946	2,4586
27	0,4712	0,4540	0,8910	0,5095	1,9626	1,1223	2,2027
30	0,5236	0,5000	0,8660	0,5774	1,7321	1,1547	2,0000
33	0,5760	0,5446	0,8387	0,6494	1,5399	1,1924	1,8361
36	0,6283	0,5878	0,8090	0,7265	1,3764	1,2361	1,7013
39	0,6807	0,6293	0,7771	0,8098	1,2349	1,2868	1,5890

continua →

Tabela 4.2: Exemplo de tabela com valores numéricos.

<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.

Ângulo		Função					
graus	rads	sen	cos	tan	cotan	sec	cosec
42	0,7330	0,6691	0,7431	0,9004	1,1106	1,3456	1,4945
45	0,7854	0,7071	0,7071	1	1	1,4142	1,4142
48	0,8378	0,7431	0,6691	1,1106	0,9004	1,4945	1,3456
51	0,8901	0,7771	0,6293	1,2349	0,8098	1,5890	1,2868
54	0,9425	0,8090	0,5878	1,3764	0,7265	1,7013	1,2361
57	0,9948	0,8387	0,5446	1,5399	0,6494	1,8361	1,1924
60	1,0472	0,8660	0,5000	1,7321	0,5774	2,0000	1,1547
63	1,0996	0,8910	0,4540	1,9626	0,5095	2,2027	1,1223
66	1,1519	0,9135	0,4067	2,2460	0,4452	2,4586	1,0946
69	1,2043	0,9336	0,3584	2,6051	0,3839	2,7904	1,0711
72	1,2566	0,9511	0,3090	3,0777	0,3249	3,2361	1,0515
75	1,3090	0,9659	0,2588	3,7321	0,2679	3,8637	1,0353
78	1,3614	0,9781	0,2079	4,7046	0,2126	4,8097	1,0223
81	1,4137	0,9877	0,1564	6,3138	0,1584	6,3925	1,0125
84	1,4661	0,9945	0,1045	9,5144	0,1051	9,5668	1,0055
87	1,5184	0,9986	0,0523	19,0811	0,0524	19,1073	1,0014
90	1,5708	1	0	-	0	-	1

**Tabela 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

Tabela 4.3: Exemplo de tabela similar a uma ficha.

## Apêndice 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.

---

**Programa A.1** Máximo divisor comum (arquivo importado).

---

```

1  FUNCTION euclid(a, b) ▷ The g.c.d. of a and b
2      r ← a mod b
3      while r ≠ 0 ▷ We have the answer if r is 0
4          a ← b
5          b ← r
6          r ← a mod b
7      end
8      return b ▷ 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>:

---

**Programa A.2** Máximo divisor comum (em português).

---

```

1  FUNCAO euclides(a, b) ▷ O máximo divisor comum de a e b
2      r ← a mod b
3  enquanto r ≠ 0 ▷ Atingimos a resposta se r é zero
4      a ← b
5      b ← r
```

*cont* →

---

<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      fim
8  devolva  $b$  ▷ O máximo divisor comum é  $b$ 
9  fim

```

---

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}}`”.



## Anexo 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 3.13.

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

Versões modernas de  $\text{\LaTeX}$  usam UTF-8, mas arquivos antigos podem usar outras codificações (como ISO-8859-1, também conhecido como `latin1` ou `Windows-1252`). Nesses casos, use `\usepackage[latin1]{inputenc}` no preâmbulo do documento. Você também pode representar os caracteres acentuados usando comandos  $\text{\LaTeX}$ : `\'a` para á, `\c{c}` para cedilha etc., independentemente da codificação usada no texto<sup>3</sup>.

- **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

<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.

<sup>3</sup> Você pode consultar os comandos desse tipo mais comuns em [en.wikibooks.org/wiki/LaTeX/Special\\_Characters](https://en.wikibooks.org/wiki/LaTeX/Special_Characters). Observe que a dica sobre o pingo do `i` não é mais válida atualmente; basta usar `\i`.

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 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.

# Referências

- [ALLCOCK 2003] William ALLCOCK. *GridFTP Protocol Specification. Global Grid Forum Recommendation (GFD.20)*. 2003 (citado na pg. 19).
- [ALVES *et al.* 2003] Carlos E. R. ALVES, Edson N. CÁCERES, Frank DEHNE e Siang W. SONG. “A parallel wavefront algorithm for efficient biological sequence comparison”. Em: *ICCSA’03: The 2003 International Conference on Computational Science and its Applications*. Springer-Verlag, mai. de 2003, pgs. 249–258 (citado na pg. 19).
- [ALVISI *et al.* 1999] Lorenzo ALVISI, Elmootazbellah ELNOZAHY, Sriram S. RAO, Syed A. HUSAIN e Asanka Del MEL. *An Analysis of Communication-Induced Checkpointing*. Rel. técn. TR-99-01. Austin, USA: Department of Computer Science, University of Texas at Austin, 1999 (citado na pg. 19).
- [BABAOGU e MARZULLO 1993] Ozalp BABAOGU e Keith MARZULLO. “Consistent global states of distributed systems: fundamental concepts and mechanisms”. Em: *Distributed Systems*. Ed. por Sape MULLENDER. 2ª ed. 1993, pgs. 55–96 (citado na pg. 19).
- [DALY 2010] Patrick W. DALY. *Reference sheet for natbib usage*. 13 de set. de 2010. URL: [mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf](http://mirrors.ctan.org/macros/latex/contrib/natbib/natnotes.pdf) (acesso em 20/12/2018) (citado na pg. 12).
- [FOWLER 2004] Martin FOWLER. *Is Design Dead?* Mai. de 2004. URL: [martinfowler.com/articles/designDead.html](http://martinfowler.com/articles/designDead.html) (acesso em 30/01/2010) (citado na pg. 19).
- [FSF 2007] FREE SOFTWARE FOUNDATION. *GNU General Public License*. 2007. URL: [www.gnu.org/copyleft/gpl.html](http://www.gnu.org/copyleft/gpl.html) (acesso em 30/01/2010) (citado na pg. 19).
- [GARCIA 2001] Islene C. GARCIA. “Visões Progressivas de Computações Distribuídas”. Tese de dout. Campinas, Brasil: Instituto de Computação, Universidade de Campinas, dez. de 2001 (citado na pg. 19).
- [KNUTH *et al.* 1996] Donald E. KNUTH, Tracy LARRABEE e Paul M. ROBERTS. *Mathematical Writing*. The Mathematical Association of America, set. de 1996 (citado na pg. 19).

- [LEHMAN *et al.* 2018] Philipp LEHMAN, Philip KIME, Moritz WEMHEUER, Audrey BORUVKA e Joseph WRIGHT. *The bibl<sub>at</sub>ex Package*. 30 de out. de 2018. URL: [mirrors.ctan.org/macros/latex/contrib/bibl<sub>at</sub>ex/doc/bibl<sub>at</sub>ex.pdf](https://mirrors.ctan.org/macros/latex/contrib/bibl<sub>at</sub>ex/doc/bibl<sub>at</sub>ex.pdf) (acesso em 20/12/2018) (citado nas pgs. 13, 19).
- [MITTELBAACH 2014] Frank MITTELBAACH. “How to influence the position of float environments like figure and table in  $\text{\TeX}$ ?” Em: *TUGboat. Communications of the  $\text{\TeX}$  Users Group* 35.3 (2014). URL: [tug.org/TUGboat/tb35-3/tb111mitt-float.pdf](https://tug.org/TUGboat/tb35-3/tb111mitt-float.pdf) (acesso em 09/01/2020) (citado nas pgs. 17, 19).
- [SCHMIDT 2003] Rodrigo M. SCHMIDT. “Coleta de Lixo para Protocolos de *Checkpointing*”. Diss. de maestr. Campinas, Brasil: Instituto de Computação, Universidade de Campinas, out. de 2003 (citado na pg. 19).

# Índice Remissivo

## B

biber, [12](#)

biblatex, [4](#), [12](#), [13](#), [19](#), [28](#)

bibtex, [4](#), [12](#), [27](#)

## F

Floats, [20](#), [21](#)

Ordem, [11](#)

## J

Java, [21](#)

## L

Legendas, [12](#), [20](#), [28](#)

## M

Mendeley, [13](#), [28](#)

Modo Matemático, [19](#)

## N

natbib, [4](#), [12](#)

Notas de rodapé, [11](#)

## S

Subfiguras, [20](#)

## Z

Zotero, [13](#), [28](#)