



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE TECNOLOGIA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E  
DE COMPUTAÇÃO



# **Avaliação de Desempenho em Redes Sem-fio Virtuais para Aplicações de IoT**

**Gilles Velleneuve Trindade Silvano**

Orientador: Prof. Dr. Ivanovich Medeiros Dantas da Silva

Co-orientador: Prof. Dr. Marcos César Madruga Alves Pinheiro

**Dissertação de Mestrado** apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Número de ordem PPgEEC: M000  
Natal, RN, julho de 2017

Divisão de Serviços Técnicos

Catálogo da publicação na fonte. UFRN / Biblioteca Central Zila Mamede

Silvano, Gilles Velleneuve Trindade.

Avaliação de Desempenho em Redes Sem-fio Virtuais para Aplicações de IoT  
/ Gilles Velleneuve Trindade Silvano - Natal, RN, 2017  
40 p.

Orientador: Ivanovich Medeiros Dantas da Silva

Co-orientador: Marcos César Madruga Alves Pinheiro

Tese (mestrado) - Universidade Federal do Rio Grande do Norte. Centro de Tecnologia. Programa de Pós-Graduação em Engenharia Elétrica e de Computação.

1. Redação técnica - Tese. 2. L<sup>A</sup>T<sub>E</sub>X- Tese. I. Melo, Sicrano Matosinho de. II. Amaral, Beltrano Catandura do. III. Título.

RN/UF/BCZM

CDU 004.932(043.2)

# **Avaliação de Desempenho em Redes Sem-fio Virtuais para Aplicações de IoT**

**Gilles Velleneuve Trindade Silvano**

Dissertação de Mestrado aprovada em 24 de julho de 2017 pela banca examinadora composta pelos seguintes membros:

---

Prof. Dr. Ivanovich Medeiros Dantas da Silva (orientador) ..... DCA/UFRN

---

Prof. Dr. Marcos César Madruga Alves Pinheiro (co-orientador) DIMAp/UFRN

---

Prof. Dr. Marcelo Henrique Ramalho Nobre ..... IFRN

---

Prof. Dr. Luiz Affonso Henderson Guedes de Oliveira ..... DCA/UFRN



*Aos meus filhos, Tico e Teco, pela  
paciência durante a realização deste  
trabalho.*



---

# Agradecimentos

---

Ao meu orientador e ao meu co-orientador, professores Ivan e Madruga, sou grato pela orientação.

Aos demais colegas de pós-graduação, pelas críticas e sugestões.

À minha família pelo apoio durante esta jornada.





---

# Resumo

---

Atualmente a Internet das Coisas (Internet of Things - IoT) é uma realidade e suas aplicações têm sido aplicadas em diferentes cenários, cada um com suas próprias restrições e requisitos. Cada nova aplicação IoT desenvolvida para um determinado cenário passa por um processo de validação e homologação antes de ser embarcada em dispositivos reais para produção. Durante o processo de validação dessas aplicações são avaliados aspectos de software, hardware e comunicação. Nesse trabalho será enfatizado aspectos de comunicação, onde dispositivos IoT majoritariamente adotam tecnologias sem fio. Nesse sentido, existem basicamente duas metodologias básicas de validação: testes de bancada reais ou simuladores de redes. Teste de bancada é o método que proporciona resultados mais acurados haja vista utilizar dispositivos reais.

Contudo, em cenários mais complexos, com alta densidade de dispositivos ou mobilidade, a validação utilizando testes de banca é bastante custosa. Por outro lado, ambientes de simulação de redes não possuem essas restrições, sendo limitados pela qualidade dos modelos matemáticos que regem o comportamento das simulações, pelos recursos computacionais disponíveis e pelo próprio simulador. Dentre os vários desafios existentes na simulação de redes, incorporar técnicas que utilizem os códigos reais dos protocolos de comunicação apresentam resultados mais próximos dos experimentos de testes de bancada. Uma solução em potencial é a simulação virtual de redes, no contexto desse trabalho, redes virtuais sem fio.

A virtualização das redes sem-fio possibilita que variáveis de ambiente possam ser manipuladas e estendidas a um custo muito baixo comparado aos testes de bancada. Portanto, este trabalho consiste em avaliar o desempenho e aplicabilidade de um simulador de redes sem-fio virtuais na validação de novas aplicações de IoT com restrições complexas.

**Palavras-chave:** Análise de Desempenho, IoT, Simuladores de Rede, Virtualização.



---

# Abstract

---

Nowadays Internet of Things (IoT) is a reality and its applications has been applied in many different scenarios, each with its own constraints and requirements. Each new IoT application developed to a specific scenario is tested and validated before being shipped on real devices for production. During the application's validation process aspects of software, hardware and communications are evaluated. In this work we will cover the aspects of communications, since IoT devices mostly adopt wireless technologies. In this way, exists basically two validation methodologies: real testbeds or network simulators. Real testbeds is the method that provides the most accurate results as it uses real devices. However, in more complex scenarios, with high density of devices or mobility, the validation process using real testbeds is very costly. On the other hand, network simulation environments doesn't have these constraints, being limited by the quality of mathematical models used during simulations, the resources available and by the simulator. Among the existing challenges in network simulations, incorporate techniques using real communication protocol's code produce more accurate results compared to real testbeds. A potential solution is the virtual network simulation, in the context of this work, virtual wireless networks.

The virtualization of wireless networks enable environment variables being manipulated and extended by a very low cost compared to real testbed methods. Thus, this work aims the performance validation and applicability of a virtual wireless network simulator in validation of new IoT applications with complex constraints.

**Keywords:** IoT, Network Simulators, Performance Analysis, Virtualization.



---

# Sumário

---

<b>Sumário</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Processando textos com $\text{\LaTeX}$ . . . . .	2
<b>Lista de Símbolos e Abreviaturas</b>	<b>1</b>
<b>2 Teoria</b>	<b>5</b>
2.1 Dimensões . . . . .	5
2.2 Divisões do documento e referências cruzadas . . . . .	6
2.3 Seções . . . . .	7
2.3.1 Subseções . . . . .	7
2.4 Índices . . . . .	7
2.4.1 Sumário . . . . .	8
2.4.2 Listas de figuras e tabelas . . . . .	8
2.4.3 Lista de símbolos (glossário) . . . . .	8
2.5 Bibliografia . . . . .	8
2.6 Considerações finais . . . . .	10
<b>3 Trabalhos relacionados</b>	<b>13</b>
3.1 Expressões matemáticas . . . . .	13
3.1.1 Equações . . . . .	15
3.1.2 Expressões multilinhas . . . . .	15
<b>4 Problema</b>	<b>17</b>
4.1 Elementos flutuantes . . . . .	17
4.1.1 Posicionamento dos elementos flutuantes . . . . .	18
4.2 Tabelas em $\text{\LaTeX}$ . . . . .	19
4.3 Figuras em $\text{\LaTeX}$ . . . . .	19
4.3.1 Imagens e fotos . . . . .	20
4.3.2 Figuras sintéticas . . . . .	21
4.4 Ferramentas para desenhos e esquemas . . . . .	24
4.5 Ferramentas para gráficos . . . . .	25

4.6	Conclusões . . . . .	25
<b>5</b>	<b>Implementação</b>	<b>27</b>
5.1	Algoritmos . . . . .	27
<b>6</b>	<b>Experimentos e Resultados</b>	<b>31</b>
<b>7</b>	<b>Conclusão</b>	<b>33</b>
7.1	Etapas de Homologação do Título . . . . .	33
7.2	Para saber mais . . . . .	34
<b>A</b>	<b>Informações adicionais</b>	<b>35</b>
	<b>Referências bibliográficas</b>	<b>35</b>

---

# Lista de Figuras

---

4.1	Trecho de <i>Os Lusíadas</i> , de Luis de Camões . . . . .	18
4.2	Exemplo de imagem real . . . . .	21
4.3	Figura vetorial grande em tamanho natural . . . . .	22
4.4	Figura <i>pixel-a-pixel</i> grande em tamanho natural . . . . .	22
4.5	Figura vetorial grande em tamanho reduzido . . . . .	22
4.6	Figura <i>pixel-a-pixel</i> grande em tamanho reduzido . . . . .	22
4.7	Figura vetorial pequena em tamanho natural . . . . .	23
4.8	Figura <i>pixel-a-pixel</i> pequena em tamanho natural . . . . .	23
4.9	Figura vetorial pequena em tamanho ampliado . . . . .	23
4.10	Figura <i>pixel-a-pixel</i> pequena em tamanho ampliado . . . . .	23
4.11	Preenchimento de figuras utilizando diferentes profundidades . . . . .	24
4.12	Imagem mesclada com elementos sintéticos . . . . .	24
4.13	Figura incluindo comandos $\LaTeX$ . . . . .	25
4.14	Exemplo de gráfico de funções matemáticas . . . . .	25





---

# Lista de Tabelas

---

3.1	Os ambientes para geração de equações multilinhas . . . . .	15
3.2	Exemplo de equação multilinha com vários pontos de alinhamento . . . . .	16
3.3	Os ambientes para geração de trechos multilinhas em equações . . . . .	16
4.1	Equação de segundo grau . . . . .	18
4.2	Raízes da equação da tabela 4.1 . . . . .	18
4.3	Tabelas com colunas de diferentes larguras e alinhamentos . . . . .	20



---

# Capítulo 1

## Introdução

---

Este documento é um modelo para propostas de tema para exames de qualificação, dissertações de mestrado e teses de doutorado a serem submetidos ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação (PPgEEC) da Universidade Federal do Rio Grande do Norte (UFRN). Procure ler o texto acompanhando o resultado produzido a partir do código fonte que o gerou.

A introdução deve dar ao leitor o posicionamento da tese e a motivação suficiente para a leitura da tese, esclarecendo:

- A natureza do problema cuja resolução se descreve (uma palhinha do problema);
- Uma indicação dos métodos usados para atacar o problema;
- As contribuições do trabalho e sua relevância para fazer progredir o estado da arte. Se for uma tese, aqui se coloca a hipótese de pesquisa, a tese, e como ela será demonstrada durante o texto;
- A forma como a tese está estruturada (estrutura do texto).

Uma dica para a montagem de uma tese ou monografia é responder as seguintes perguntas, cada uma em um parágrafo:

- **O que fez?:** O que se propõe, o que se trabalhou, “Propomos...”;
- **Como fez?:** Técnicas, metodologia de trabalho, “Para tal...”;
- **Por que?:** Quais as motivações e justificativas do trabalho. A motivação inclui se faz parte de um projeto maior, o fato do problema não ter solução, e/ou qual a história que o levou a trabalhar nisso. A justificativa não é obrigatória, mas pode motivar o leitor;
- **Quanto vale?:** Quais as contribuições do trabalho. Inclui hipótese, tese, e resultados principais (técnicas, métodos novos, produtos, publicações);
- **Para quem?** Quais as aplicações que se beneficiarão do seu trabalho, para quê será (ou poderá ser) usado - como parte de um projeto;
- **Como está?** Estrutura do texto. Evite escrever coisas óbvias (como “na Introdução, introduzimos...”, “na conclusão, concluímos”). Coloque texto informativo, que acrescente o saber de quem lê (como “na seção 2 será mostrado por que **wavelets** podem ser usadas para estabilizar corrente”).

Uma observação importante é não colocar “Objetivos” nos trabalhos finais, pois não são mais objetivos, uma vez que você os atingiu. Neste caso, estes viram contribuições.

Os “Objetivos” são mais utilizados para qualificações e projetos de pesquisa, que ainda não foi desenvolvidos. Tomar cuidado também para colocar objetivos factíveis (porque será um problema se não os cumprir...).

## 1.1 Processando textos com L<sup>A</sup>T<sub>E</sub>X

O L<sup>A</sup>T<sub>E</sub>X não é um editor de texto no sentido convencional. Ele funciona muito mais como um “compilador” de uma linguagem de programação de textos:

1. Inicialmente você escreve um “programa” nesta linguagem de programação de textos (linguagem L<sup>A</sup>T<sub>E</sub>X), dizendo o conteúdo e a formatação do seu documento. Para escrever este “programa” pode-se usar qualquer editor de textos capaz de salvar documentos em formato texto ASCII puro.
2. O próximo passo é compilar este código fonte, produzindo um arquivo `.dvi` que funciona como um “programa objeto”. Esta compilação é feita pelo programa `latex`.
3. Em seguida o arquivo `.dvi` é convertido para o formato no qual se deseja produzir o texto: PostScript (usando o programa `dvips`), PDF (usando o `pdflatex`, que já faz esta fase e a fase anterior) ou visualização na tela (`xdvi`).

Estas explicações tomaram como base a versão do L<sup>A</sup>T<sub>E</sub>X mais comum para sistemas operacionais Unix. Existem outras implementações tanto para Unix quanto para Windows, onde os comandos executados são diferentes mas a idéia geral é sempre a mesma.

Além do `latex` para compilar o texto, pode ser necessário executar outros programas, como o `bibtex` para incluir automaticamente as referências bibliográficas ou o `makeindex` para gerar o glossário. Estes programas devem ser chamados em uma ordem específica. Para automatizar este processo, é fornecido um arquivo `Makefile`, de modo que a compilação completa pode ser feita utilizando um dos seguintes comandos:

`make`: executa a tarefa *par default*, que pode ser alterada no `Makefile` para apontar para qualquer uma das seguintes.

`make simples`: apenas executa o `latex` uma vez;

`make principal.dvi`: executa todos os passos e aplicativos necessários para produzir o arquivo `.dvi` completo;

`make principal.ps`: executa todos os passos e aplicativos necessários para produzir o arquivo PostScript `principal.ps` completo;

`make principal.pdf`: executa todos os passos e aplicativos necessários para produzir o arquivo PDF `principal.pdf` completo;

`make clean`: remove todos os arquivos intermediários gerados no processo de compilação, inclusive o `principal.dvi`.

`make realclean`: além de fazer um `make clean`, remove os arquivos `principal.ps` e `principal.pdf`.

Neste documento, além de dicas sobre como montar cada capítulo, serão apresentadas informações importantes e dicas sobre como utilizar o L<sup>A</sup>T<sub>E</sub>X neste tipo de documento.

O capítulo 2 apresenta as diretrizes gerais sobre a formatação dos textos. O capítulo 3 apresenta alguns recursos do L<sup>A</sup>T<sub>E</sub>X para escrever expressões matemáticas, enquanto o capítulo 4 trata da inclusão de tabelas, gráficos e figuras no documento. Um recurso para a construção de algoritmos pode ser visto em 5. Por fim, o capítulo 7 mostra alguns exemplos de construção automática de bibliografias utilizando o aplicativo B<sub>B</sub>L<sub>A</sub>T<sub>E</sub>X e menciona fontes adicionais para mais informações.



---

# Capítulo 2

## Teoria

---

A teoria (ou referencial teórico) deve descrever as técnicas e metodologias existentes e desenvolvimentos anteriores que sejam essenciais ao entendimento do trabalho, sempre citando referencial teórico para cada técnica abordada. Aqui, as ferramentas matemáticas, mesmo que já conhecidas, podem ser revisitadas com as devidas citações. Geralmente, estas informações são obtidas em livros, *survey papers* ou *seminar papers*. É importante lembrar que o trabalho relatado em uma tese ou dissertação, muito mais do que em um artigo, deve ser auto-contido e reproduzível.

Além de instruções sobre a montagem da teoria, este capítulo também apresenta considerações de ordem geral sobre a organização que deve ser adotada no seu documento, tais como número de páginas, margens e subdivisões.

### 2.1 Dimensões

Não há um número mínimo ou máximo de páginas para propostas de tema, dissertações ou teses. Entretanto, se o seu documento for muito menor do que a média pode transmitir uma ideia de falta de conteúdo a apresentar. Por outro lado, um documento muito grande corre o risco de só conseguir a atenção total do leitor no seu início, fazendo com que as partes mais importantes, que geralmente estão no final do documento, não sejam devidamente consideradas. Apenas para servir como parâmetro, estão indicados a seguir os limites usuais quanto ao número de páginas<sup>1</sup> dos documentos do PPgEEC da UFRN, adotando as margens e os espaçamentos definidos neste modelo:

- Proposta de tema para exame de qualificação de mestrado: entre 20 e 40 páginas
- Proposta de tema para exame de qualificação de doutorado: entre 30 e 50 páginas
- Dissertação de mestrado: entre 50 e 100 páginas
- Tese de doutorado: entre 80 e 150 páginas

O tamanho padrão para a fonte é de 12pt. Para facilitar a escrita de comentários, sugestões e correções da banca, recomenda-se o espaçamento 1.5 entre as linhas do texto e a impressão em um único lado da folhas para os seguintes documentos:

---

<sup>1</sup>Uma folha corresponde a uma página em impressão em face simples e a duas páginas em impressão em face dupla

- Proposta de tema para exame de qualificação;
- Versão inicial de dissertação de mestrado; e
- Versão inicial de tese de doutorado.

Para as versões finais de teses e dissertações, onde se busca uma melhor qualidade visual e tipográfica do texto, deve-se utilizar espaçamento simples entre as linhas e a impressão nos dois lados da página.

As margens devem seguir os valores adotados neste documento, que podem ser verificados no arquivo `principal.tex`. É importante notar que, na versão final de teses ou dissertações, recomenda-se a impressão nos dois lados da página. Por esta razão, a margem direita em páginas pares deve ter o mesmo valor que a margem esquerda em páginas ímpares e vice-versa, para que a encadernação fique correta. Também em razão da impressão em frente e verso, os capítulos devem sempre começar em uma página de número ímpar, com a eventual inclusão de uma página em branco. O  $\text{\LaTeX}$  se encarrega de fazer automaticamente estes ajustes.

## 2.2 Divisões do documento e referências cruzadas

Documentos do porte de uma tese ou dissertação devem ser subdivididos em capítulos. O capítulo deve conter uma introdução e um fecho.

A introdução do capítulo fornece ao leitor uma breve descrição do que será tratado no capítulo e não forma uma seção: para exemplificar, a introdução deste capítulo é o parágrafo que precede a primeira seção.

O fecho do capítulo apresenta comentários finais sobre o que foi desenvolvido no capítulo e/ou faz uma ligação com o que será visto no capítulo seguinte; normalmente é colocado em uma seção específica, denominada “Comentários Finais”, “Conclusões”, “Resultados”, “Avaliação Final” ou qualquer outra denominação que se adéque ao texto.

Capítulos são divididos em seções. O número ideal de seções é impossível de se precisar. Entretanto, um capítulo com uma única seção provavelmente deve ser agregado ao capítulo anterior ou posterior. Um capítulo com quinze seções provavelmente deve ser subdividido em dois capítulos.

Capítulos, seções e subseções devem ser rotulados para que possam ser referenciados em qualquer parte do texto. Isto é feito com o comando `\label{}`, que deve ser colocado logo após (nunca antes) o comando que criou a seção, capítulo, etc. O parâmetro do comando `label` é o nome simbólico que será usado para se fazer referência a esta entidade dentro do texto, com o comando `\ref{}`. O nome pode ser qualquer coisa, mas não pode conter acentos, por exemplo. Neste documento nós utilizamos a convenção de prefixar os rótulos dos capítulos com `Cap:`, das seções com `Sec:`, das equações com `Eq:` e assim por diante, mas esta convenção não é obrigatória. Veja a seguir um exemplo de utilização das referências cruzadas:

... no capítulo 1 apresentamos um modelo de capítulo de tese.

Note que, no código fonte deste trecho de frase, o espaço entre a palavra `capítulo` e o comando `\ref{}` foi escrito com um `~` e não com um espaço normal. O `~` é o comando



$\text{\LaTeX}$  para criar um espaço onde não se pode mudar de linha, pois ficaria estranho se o texto “no capítulo” estivesse no fim de uma linha e o número 1 no início da outra linha.

Existe uma particularidade no código fonte do parágrafo anterior. Para se escrever:

...o comando  $\text{\LaTeX}$  para criar...

se colocou depois do comando  $\text{\LaTeX}$  um espaço precedido de uma contrabarra, ao invés de um espaço normal. Isto porque espaços depois de comandos são ignorados pelo  $\text{\LaTeX}$ ; com um espaço normal as palavras ficariam ligadas:

...o comando  $\text{\LaTeX}$  para criar...

Ao invés do espaço precedido pela contrabarra, poder-se-ia também utilizar um  $\sim$ . A diferença é que neste caso o  $\text{\LaTeX}$  não poderia fazer uma quebra de linha entre as palavras.

## 2.3 Seções

Seções são divisões do conteúdo do capítulo. Esta divisão deve ser lógica (temática) e não física (por tamanho). Por exemplo, um capítulo que trata de *software* de sistema teria seções que tratam de montadores, ligadores, carregadores, compiladores e sistemas operacionais.

Tal como capítulos, seções devem ser rotuladas para referência em outras partes do texto. Seções são divididas em subseções.

### 2.3.1 Subseções

Subseções são divisões de seções. No exemplo do texto sobre *software* de sistema, a seção referente a sistema operacional conteria, por exemplo, subseções que tratam de arquivos, processos, memória e entrada/saída. Tal como seções, subseções são divisões temáticas do texto.

#### Subsubseções

Subsubseções são divisões de subseções e não devem ser numeradas no texto. O  $\ast$  após o comando `subsubsection*` instrui o  $\text{\LaTeX}$  a não numerar a subsubseção. Esta mesma regra se aplica a outros comandos. Por exemplo, `\chapter{}` inicia um capítulo, enquanto `\chapter*{}` inicia um capítulo sem número. O comando `chapter*` foi usado no arquivo `resumos.tex` para criar os capítulos não numerados referentes ao resumo e ao *abstract*.

## 2.4 Índices

O  $\text{\LaTeX}$  é capaz de gerar automaticamente o índice do texto (sumário), os índices de figuras e de tabelas e uma lista de símbolos ou glossário.

### 2.4.1 Sumário

Todas as divisões numeradas (capítulos, seções e subseções) são automaticamente incluídas no sumário. Ao se criar uma nova divisão é necessário compilar duas vezes o texto com o  $\text{\LaTeX}$ : na primeira compilação será percebida a inclusão da nova divisão, enquanto na segunda será gerado o índice atualizado. Esta mesma necessidade de uma dupla compilação aparece quando se acrescenta qualquer nova referência cruzada: uma nova figura ou tabela, uma nova referência bibliográfica, etc.

Além das divisões que são incluídas automaticamente no sumário, pode-se incluir manualmente outras informações. Os índices, por exemplo, não são incluídos automaticamente no sumário. Verifique no arquivo `principal.tex` o que deve ser feito para fazer esta inclusão.

### 2.4.2 Listas de figuras e tabelas

Estas listas são geradas automaticamente a partir dos `caption's` de todos os ambientes `figure` e `table`. Maiores detalhes sobre estes ambientes serão apresentados no capítulo 4.

### 2.4.3 Lista de símbolos (glossário)

Este ambiente pode ser utilizado para produzir uma lista de símbolos, um glossário ou uma lista de abreviaturas. Ao utilizar pela primeira vez uma entidade que precise de definição, o autor, ao final do parágrafo, gera a entrada para o glossário. A título de exemplo, foram incluídos na lista alguns símbolos e abreviaturas que aparecem no texto a seguir:

As primeiras teses apresentadas no PPgEEC da UFRN eram datilografadas manualmente. Para escrever uma fórmula simples como:

$$\omega = \omega_0 + \alpha \cdot \Delta t \quad (2.1)$$

os autores tinham que desenhar os símbolos manualmente ou, para os mais afortunados, trocar inúmeras vezes a esfera da sua máquina de datilografia.

A ordem de aparição dos verbetes no glossário é a seguinte: inicialmente os símbolos, depois os números e por último as *strings*. Você pode modificar esta ordem incluindo um parâmetro adicional no comando `\nomenclature[opcional]{simb}{signif}`. No exemplo, este parâmetro foi utilizado para colocar UFRN antes de PPgEEC.

## 2.5 Bibliografia

As referências bibliográficas são incluídas dentro de um ambiente específico para este fim, através dos comandos `\begin{thebibliography}` e `\end{thebibliography}`. Um documento só pode conter um único destes ambientes. A referência aos documentos

no texto é feita usando-se referências cruzadas e chaves simbólicas, da mesma forma que as equações e as figuras. Cada documento dentro de um ambiente `thebibliography` é introduzido por um comando `\bibitem`. O argumento obrigatório (entre chaves) do comando `\bibitem` é a chave simbólica pela qual o documento será citado no texto, usando o comando `\cite`. O argumento opcional (entre colchetes) do comando `\bibitem` é a expressão que será inserida tanto no texto, no local onde a referência foi citada, quanto na lista de referências bibliográficas, como etiqueta do documento em questão.

O ambiente `thebibliography` pode ser digitado diretamente pelo usuário ou gerado automaticamente a partir de um arquivo de informações bibliográficas. A digitação manual tem a vantagem de tornar o documento autocontido, enquanto a geração automática permite um melhor reaproveitamento das informações e uma maior uniformidade das referências nos diversos documentos. Sempre que possível, aconselha-se a geração automática, que é feita pelo aplicativo `BIBTEX`.

A principal vantagem da geração automática de bibliografias é que se pode manter um arquivo único com todas as referências bibliográficas que foram ou podem vir a ser usadas em algum dos seus documentos (artigos, tese, etc.). O `BIBTEX` se encarrega de verificar quais delas foram efetivamente citadas no documento sendo processado e gerar um ambiente `thebibliography` que contém apenas os documentos necessários.

As informações bibliográficas devem ser salvas em um arquivo no formato `BIBTEX` e com extensão `.bib`. O formato `BIBTEX` permite referenciar diferentes tipos de documentos:

- artigos em revistas (?);
- artigos em anais de simpósios (?);
- artigos em coletâneas de artigos (?);
- capítulos de livros (?);
- anais de simpósios (?);
- livros (?);
- teses de doutorado (?);
- teses de mestrado (?);
- relatórios técnicos (?);
- manuais técnicos (?);
- trabalhos não publicados (?);
- páginas na Internet (?) (a data é o dia do último acesso à página);
- miscelânea (?).

O arquivo `BIBTEX` não contém nenhuma informação de formatação. Esta formatação é definida através do comando `\bibliographystyle`. Este modelo inclui um arquivo de estilo (`ppgee.bst`) que gera as referências no padrão adotado para os documentos do PPgEEC. Este estilo é baseado no padrão `jmr` do pacote `harvard`, com as modificações necessárias para a língua portuguesa. Para maiores informações, veja a documentação do pacote `harvard`, disponível na Internet ou na maiorias das instalações `LATEX`.

A lista de referências bibliográficas é gerada pelo comando `\bibliography`, cujo parâmetro obrigatório é o nome do(s) arquivo(s) que contém(êm) as informações bibliográficas. O `BIBTEX` se encarrega de extrair deste(s) arquivo(s) as referências citadas,

formatá-las de acordo com o estilo escolhido e gerar o ambiente `thebibliography` correspondente. Este ambiente é salvo em um arquivo com terminação `.bbl`, que é automaticamente inserido no documento no local do comando `\bibliography`. Este procedimento pode ser melhor compreendido analisando-se os arquivos `principal.tex` e `bibliografia.bib`, além do arquivo `principal.bbl` gerado automaticamente pelo BIB T<sub>E</sub>X.

Uma recomendação importante é que as citações não fazem parte do texto; portanto, as frases devem fazer sentido mesmo que as expressões de citação sejam removidas. Para exemplificar, não se deve usar:

...conforme demonstrado por (?).

e sim:

...conforme demonstrado por ? (para tal citação, utiliza-se o comando `\citeasnoun`).

...conforme demonstrado na literatura (?, e.g. ).<sup>2</sup>

## 2.6 Considerações finais

O simples fatos de utilizar corretamente uma boa ferramenta de formatação de textos e de seguir as recomendações quanto ao estilo não garantem a qualidade do documento produzido. O principal aspecto a ser levado em conta é a qualidade da redação.

Propostas de tema, dissertações e teses devem ser escritas em linguagem técnica, que difere em alguns aspectos da linguagem literária. Em textos das ciências exatas e tecnológicas, o objetivo principal é a clareza e a correção, algumas vezes com um certo prejuízo da estética literária. Algumas recomendações quanto à redação são as seguintes:

- Evite períodos longos, reduzindo o número de apostos, orações subordinadas, pronomes relativos (o qual, que, cujo, o mesmo, do qual, etc.) e inversões na ordem normal de aparecimento dos elementos da frase (sujeito, verbo, predicado). Divida períodos longos em várias frases menores, mesmo que seja necessário repetir alguns termos. Por exemplo:

*Esta versão foi concebida por ocasião da disciplina de Sistemas de Transmissão de Dados, que tinha como principal objetivo, fazer uso de uma plataforma móvel onde pudesse ser aplicada uma técnica de transmissão de dados, possibilitando dessa forma o controle desta.*

pode ser substituído por:

*Esta versão foi concebida durante a disciplina de Sistemas de Transmissão de Dados. O principal objetivo desta primeira versão era aplicar uma técnica de transmissão de dados a uma plataforma móvel. O uso da técnica de transmissão de dados tornou possível o controle da plataforma móvel.*

---

<sup>2</sup>A abreviatura e.g. significa por exemplo. Vem do latim *exempli gratia*. Também se usa, para o mesmo caso, v.g. (*verbi gratia*) ou simplesmente p.ex.

- Verifique a pontuação empregada para não separar por vírgulas o sujeito do verbo, como no exemplo anterior em “...*tinha como principal objetivo, fazer uso de...*”, nem trocar pontos por vírgulas ou vice-versa. Estes erros geralmente aparecem associados a períodos muito longos, como nos dois exemplos a seguir:

*Quando a operação é realizada sem o sistema de recepção, o veículo se desloca e realiza tarefas, baseadas na programação do dispositivo microcontrolador e leitura de sensores, como a odometria, por exemplo, a tabela abaixo lista a função atribuída a cada pino.*

*A garra é dotada também de um conjunto de sensores mecânicos de fim de curso tipo contato seco normalmente aberto (NA), nas posições de máxima elevação e abertura, bem como dois sensores desse mesmo tipo nas pinças da garra que respondem a uma pequena pressão em sua superfície, sendo responsáveis por pressionar o elemento objeto que se deseja recolher. Evitando que o mesmo escorregue quando erguido.*

- Evite o uso de adjetivos que expressam julgamentos de valor de maneira não quantificável. Por exemplo, ao invés de dizer que os resultados foram excelentes, diga que em 98% dos experimentos o erro foi menor que 1%, deixando ao leitor a tarefa de concluir se estes resultados são excelentes ou não.

No que diz respeito aos recursos de formatação, lembre-se que o  $\text{\LaTeX}$  já faz a maior parte do trabalho para você, seguindo padrões que foram definidos por especialistas para garantir uma boa qualidade tipográfica. Portanto, tente não modificar “manualmente” a formatação gerada. Para isto, evite sempre que possível os seguintes recursos, pois um texto “limpo” é mais agradável de ler que um texto “enfeitado”:

- letras maiúsculas em palavras onde elas não têm justificativa gramatical: nomes de meses (janeiro e não Janeiro), apenas para dar ênfase (... foi desenvolvido um Robô Móvel com rodas...), etc.
- **o uso indiscriminado de negrito;**
- *o uso de itálico, exceto em palavras em outra língua ou em definições de termos que aparecem pela primeira vez;*
- texto em fontes diferentes: espaçamento uniforme, **sem serifa**, CAIXA ALTA, etc.
- o uso de texto sublinhado;
- o uso excessivo de notas de rodapé<sup>3</sup>.
- palavras em língua estrangeira quando existe uma equivalente em português (desempenho e não *performance*) ou neologismos não justificados (downloadar, randômico, etc.)

---

<sup>3</sup>Notas de rodapé são estas expressões no pé da página.



---

## Capítulo 3

### Trabalhos relacionados

---

Nos trabalhos relacionados, coloque a essência dos trabalhos no domínio abordado (com referências bibliográficas). Isto não inclui citações específicas e particulares para trabalhos anteriores (seus e de outros), mas sim, algo mais genérico, por categoria ou características (por exemplo, classificar os trabalhos segundo as técnicas empregadas). É muito interessante também construir uma tabela (quem lê gosta muito e com certeza elogia!). Escolha apenas os trabalhos estritamente relacionados com o seu (muito similares) para comentar especificamente, sem categorias (por exemplo, “O trabalho de Fulano [10] propõe uma abordagem muito similar à nossa, porém diferindo pela base wavelet usada...”). Ao final, contextualize o seu trabalho, situando-o em relação ao quadro (tabela) e a esses estritamente relacionados (dizer no que o seu trabalho melhora ou se distingue dos anteriores). E, principalmente, lembre-se de ser educado sempre: nunca desmereça os trabalhos dos outros.

#### 3.1 Expressões matemáticas

L<sup>A</sup>T<sub>E</sub>X é insuperável no processamento de expressões matemáticas. Expressões simples como  $2^n$  podem ser editadas no próprio texto. Equações complexas como:

$$p(\gamma) = \frac{1}{2} \sqrt{\frac{M}{\tilde{\gamma}_b}} \frac{1}{\prod_{i=1}^M \sqrt{\tilde{\gamma}_i}} \int_0^{\sqrt{M\delta}} \int_0^{\sqrt{M\delta}-r_M} \dots \int_0^{\sqrt{M\delta}-\sum_{i=3}^M r_i} p\left(\frac{\sqrt{M\delta}-\sum_{i=2}^M r_i}{\sqrt{\tilde{\gamma}_1}}, \frac{r_2}{\sqrt{\tilde{\gamma}_2}}, \dots, \frac{r_M}{\sqrt{\tilde{\gamma}_M}}\right) dr_2 \dots dr_{M-1} dr_M \quad (3.1)$$

ou:

$$T(r) = \frac{1}{f_m} \left( \frac{\pi}{2} \sum_{i=1}^M \tilde{r}_i^2 \dot{\tilde{\varsigma}}_i^2 \right)^{-1/2} \frac{\int_0^{\rho\sqrt{M}} \int_0^{\rho\sqrt{M}-r_M} \dots \int_0^{\rho\sqrt{M}-\sum_{i=3}^M r_i} \int_0^{\rho\sqrt{M}-\sum_{i=2}^M r_i} p\left(\frac{r_1}{\tilde{r}_1}, \frac{r_2}{\tilde{r}_2}, \dots, \frac{r_M}{\tilde{r}_M}\right) dr_1 dr_2 \dots dr_{M-1} dr_M}{\int_0^{\rho\sqrt{M}} \int_0^{\rho\sqrt{M}-r_M} \dots \int_0^{\rho\sqrt{M}-\sum_{i=3}^M r_i} p\left(\frac{\rho\sqrt{M}-\sum_{i=2}^M r_i}{\tilde{r}_1}, \frac{r_2}{\tilde{r}_2}, \dots, \frac{r_M}{\tilde{r}_M}\right) dr_2 \dots dr_{M-1} dr_M} \quad (3.2)$$

são automaticamente numeradas e podem ser referenciadas a partir do texto. Por exemplo, a equação 3.2 é trivialmente derivada da equação 3.1.

No parágrafo anterior foi intencionalmente introduzido um erro. Note que o trecho de frase “são automaticamente numeradas...” tem uma indentação, ou seja, um espaço inicial de tabulação, como se fosse a primeira frase de um novo parágrafo. Na realidade, todo o texto anterior constitui um único parágrafo, no meio do qual se inserem as duas equações. Por esta razão, os trechos de frase após as equações não devem se iniciar com letra maiúscula nem ser indentados. É importante lembrar que nestas situações em que frases são interrompidas por equações é obrigatória a inclusão de dois pontos no fim dos trechos da frase, como em “...Equações complexas como:” e em “ou:”.

O que gerou este erro de indentação? Lembre-se que em  $\text{\LaTeX}$  uma linha em branco no código fonte indica a separação entre dois parágrafos. A causa do problema é a linha em branco entre o `\end{equation}` e o `são automaticamente numeradas...`. Como regra geral, enquanto um parágrafo não for encerrado, não podem ser incluídas linhas em branco, mesmo que no meio do parágrafo existam equações, figuras, notas de rodapé, etc.

Pequenas expressões matemáticas como  $x_0^2$  podem ser inseridas diretamente no texto, delimitadas por cifrões (\$). Deve-se evitar este recurso com expressões muito grandes, como  $\begin{bmatrix} 1 & \frac{2}{x+1} \\ -2 & 1 \end{bmatrix}^{-1}$ , porque o espaçamento entre as linhas fica prejudicado. Para incluir expressões não numeradas maiores, pode-se utilizar o par de delimitadores `\[ e \]`, o que gera expressões centralizadas na página:

$$\left( \begin{array}{cc} 1 & \frac{2}{x+1} \\ -2 & 1 \end{array} \right)^{-1} = \left( \begin{array}{cc} \frac{x+1}{x+5} & -\frac{2}{x+5} \\ \frac{2(x+1)}{x+5} & \frac{x+1}{x+5} \end{array} \right) \quad \text{se } x \neq -1 \quad \text{e } x \neq -5$$

Um erro comum em expressões matemáticas é o de digitar os nomes de funções diretamente, sem utilizar os comandos apropriados. Por exemplo, a expressão  $\sin(\omega t + \phi)$  está correta, enquanto que a expressão  $\sin(\omega t + \phi)$  é interpretada pelo  $\text{\LaTeX}$  como sendo o produto das variáveis  $s$ ,  $i$  e  $n$  pela expressão entre parênteses. Para as funções usuais já existem comandos predefinidos, como `\sin`. Para suas próprias funções, utilize o comando `\operatorname{}`, diretamente ou definindo um novo comando:

$$\text{fatorial}(x) = x \cdot \text{fatorial}(x - 1)$$

O  $\text{\LaTeX}$  possui uma sintaxe específica para índices (sub-escritos) e expoentes (super-escritos) posicionados do lado direito do objeto a que se referem, mas não do lado esquerdo. Para conseguir este efeito, adicione índices e/ou expoentes a um bloco vazio posicionado antes do objeto:

$$\begin{array}{ccc} w^2 & w_i & {}^z w \\ {}^* w & {}_p w^Q & {}_B^A w_D^C \end{array}$$



### 3.1.1 Equações

As equações são delimitadas por `\begin{equation}` e `\end{equation}`. Devem ser identificadas por um `\label` para permitir referências futuras:

$$y = g(x) = g(x_{PO}) + \left. \frac{dg}{dx} \right|_{x=x_{PO}} \frac{(x-x_{PO})}{1!} + \left. \frac{d^2g}{dx^2} \right|_{x=x_{PO}} \frac{(x-x_{PO})^2}{2!} + \dots \quad (3.3)$$

Devem-se evitar referências com expressões do tipo “a equação acima” e “a próxima equação”, pois modificações no texto podem tornar a referência inválida. Use sempre referências pelo rótulo (`\label`), como em “a equação 3.3”, mesmo para equações próximas.

Existem vários ambientes para escrever equações, como o `cases` para construir expressões condicionais:

$$f(x) = 1 + \begin{cases} 0 & \text{se } x = 0 \\ 1/x & \text{caso contrário} \end{cases} + \begin{cases} x/2 & \text{se } x \text{ é inteiro e par} \\ \frac{x+1}{2} & \text{se } x \text{ é inteiro e ímpar} \\ \frac{x+0.5}{2} & \text{se } x \text{ não é inteiro} \end{cases} \quad (3.4)$$

### 3.1.2 Expressões multilinhas

O pacote `amstex` define vários ambientes para criar expressões matemáticas que ocupam mais de uma linha. Existem versões dos ambientes com e sem inclusão do número na equação, conforme indicado na tabela 3.1<sup>1</sup>.

PACOTE		Tipo de alinhamento
Com número	Sem número	
<code>gather</code>	<code>gather*</code>	sem alinhamento (só múltiplas linhas)
<code>multiline</code>	<code>multline*</code>	quebra de equação em várias linhas
<code>align</code>	<code>align*</code>	alinhamento em um único ponto
<code>alignat</code>	<code>alignat*</code>	alinhamento em vários pontos, no centro da linha
<code>xalignat</code>	<code>xalignat*</code>	vários pontos, ocupando toda a linha (com margens)
<code>-</code>	<code>xxalignat</code>	vários pontos, ocupando toda a linha (sem margens)

Tabela 3.1: Os ambientes para geração de equações multilinhas

Para os ambientes `alignat`, `xalignat` e `xxalignat` existe um parâmetro obrigatório que é o número de elementos em cada linha. Cada elemento tem um ponto de alinhamento com os outros elementos da mesma coluna em outras linhas. Cada elemento é separado do próximo por um `&`. Dentro de cada elemento, um novo `&` marca seu ponto de alinhamento, conforme o exemplo da tabela 3.2, onde se vê o código fonte e o resultado produzido.

Os ambientes da tabela 3.1 funcionam como um ambiente `equation`: ocupam toda a linha e centralizam a expressão. Em algumas situações, entretanto, se deseja incluir um sub-ambiente multilinhas dentro de uma expressão mais geral. Para isto, o `amstex` fornece as opções listadas na tabela 3.3<sup>1</sup>.

<sup>1</sup>Nestas tabelas foram utilizadas extensões do ambiente `tabular` fornecidas pelo pacote `tabularx`. Esta

---

<pre>\begin{xxalignat}{3} &amp;aaaa &amp; mm&amp;mm &amp; cccc&amp; \\ &amp;xxxx&amp; &amp; ii&amp;ii &amp; llll&amp; \end{xxalignat}</pre>	$\Rightarrow$	$\begin{array}{ccc} aaaa & mmmm & cccc \\ xxx & iii & lll \end{array}$
---	---------------	--

---

Tabela 3.2: Exemplo de equação multilinha com vários pontos de alinhamento

PACOTE	Tipo de alinhamento
gathered	sem alinhamento (só múltiplas linhas)
aligned	alinhamento em um único ponto
alignedat	alinhamento em vários pontos

Tabela 3.3: Os ambientes para geração de trechos multilinhas em equações

A equação 3.5 ilustra a utilização de ambientes `aligned` inseridos dentro de um ambiente `alignat`. Nos ambientes multilinhas numerados cada linha terá seu próprio número e deverá receber seu próprio rótulo (`label`), exceto caso se informe explicitamente que a linha não deverá ser numerada, usando o comando `\nonumber`.

$$\begin{cases} \dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y} = \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{cases} \Rightarrow \begin{cases} s\mathbf{X}(s) - \mathbf{x}(0) = \mathbf{A}\mathbf{X}(s) + \mathbf{B}\mathbf{U}(s) \\ \mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \end{cases} \Rightarrow$$

$$\begin{cases} \mathbf{X}(s) = \Phi(s)\mathbf{x}(0) + \Phi(s)\mathbf{B}\mathbf{U}(s) \\ \mathbf{Y}(s) = \mathbf{C}\mathbf{X}(s) + \mathbf{D}\mathbf{U}(s) \end{cases} \quad \text{onde} \quad \Phi(s) = (s\mathbf{I} - \mathbf{A})^{-1} \quad (3.5)$$

---

# Capítulo 4

## Problema

---

Aqui você deve especificar o seu problema (de forma muito formal). Coloque a matemática (ou descreva a problemática) do problema. Use e abuse de Definições, Teoremas, Proposições e Provas. Descreva a sua solução (matemática) para o problema em questão, como você resolve o problema. Pode ser descritivo, mas provando que funciona matematicamente (ou formalmente).

### 4.1 Elementos flutuantes

Uma das maiores dificuldades na edição de textos de qualidade é o posicionamento dos elementos gráficos: figuras, gráficos e tabelas. Como estes elementos muitas vezes são grandes, aparece o dilema sobre o que fazer quando uma quebra de página deveria acontecer no meio do elemento. Há duas possibilidades:

1. O autor informa exatamente onde o elemento gráfico deve ficar no texto, evitando que quebras de páginas aconteçam no meio de um elemento. O problema com esta abordagem é que todo o trabalho de posicionamento pode ser perdido caso se inclua ou se exclua algum texto ou elemento.
2. O editor de texto posiciona os elementos gráficos de forma a não deixar espaços em branco nas páginas. Estes elementos que podem ser posicionados pelo editor são conhecidos como *elementos flutuantes*. O problema com esta abordagem é que o posicionamento adotado pode não corresponder às expectativas do autor.

O  $\text{\LaTeX}$  oferece as duas possibilidades de posicionamento. Este capítulo apresenta exemplos de inclusão de elementos gráficos no texto, bem como algumas ferramentas externas ao  $\text{\LaTeX}$  que podem ser utilizadas para gerá-los.

Para caracterizar uma parte do texto como sendo flutuante, ela deve ser delimitada por `\begin{figure}` e `\end{figure}` ou por `\begin{table}` e `\end{table}`. Apesar do que os nomes sugerem, nada obriga que o ambiente `figure` seja usado para delimitar figuras ou que o ambiente `table` seja usado para delimitar tabelas, embora esta seja a escolha quase sempre adotada. Estes dois ambientes são praticamente equivalentes, com as seguintes diferenças:

- os dois ambientes usam contadores diferentes para numerar os elementos flutuantes;

Figura 4.1: Trecho de *Os Lusíadas*, de Luis de Camões

As armas e os barões assinalados	E também as memórias gloriosas
Que da ocidental praia lusitana	Daqueles reis que foram dilatando
Por mares nunca dantes navegados	A Fé, o Império, as terras viciosas
Passaram ainda além da Trapobana	De África e Ásia andaram devastando,
Em perigos e guerras esforçados	E aqueles que por obras valerosas
Mais do que prometia a força humana	Se vão da lei da morte libertando:
Entre gente remota edificaram	Cantando espalharei por toda parte,
Novo reino, que tanto sublimaram	Se a tanto me ajudar o engenho e arte.

- os ambientes `figure` serão incluídos na `listoffigures`, enquanto os ambientes `table` serão incluídos na `listoftables`;
- as legendas (`caption's`) dos ambientes `figure` serão precedidas da palavra “Figura ...”, enquanto as legendas dos ambientes `table` serão precedidas da palavra “Tabela ...”. Estas duas palavras podem ser alteradas pelo autor.

Para ilustrar o fato de que estes ambientes podem conter virtualmente qualquer coisa, a figura 4.1 contém um texto que foi tornado flutuante por ser incluído em um ambiente `figure` e as tabelas 4.1 e 4.2 contêm expressões matemáticas flutuantes, incluídas em um ambiente `table`. A tabela (table) 3.3 na página 16 também não contém uma tabela no sentido estrito do termo, mas sim uma linha de texto formada por duas `minipage's` separadas por um espaço horizontal. A primeira `minipage` contém um trecho de código fonte e a segunda, o resultado produzido (uma expressão matemática multialinhada).

É importante ressaltar que o que é numerado é o `caption` e não a `figure` ou a `table`. Portanto, o `label` deve ser colocado sempre após o `caption` ao qual ele se refere. Conforme ilustram as tabelas 4.1 e 4.2, uma mesma `figure` ou `table` pode ter mais de um ou nenhum `caption`. O `caption` pode ser colocado antes do conteúdo flutuante, como na figura 4.1, ou depois, como nas tabelas 4.1 e 4.2. Nos documentos do PPgEEC, o padrão é sempre posicionar o `caption` abaixo das figuras e das tabelas.

### 4.1.1 Posicionamento dos elementos flutuantes

Em cada `\begin{figure}` ou `\begin{table}` pode-se incluir um parâmetro opcional com as opções de posicionamento para este elemento flutuante. Parâmetros adicionais de comandos  $\text{\LaTeX}$  são sempre fornecidos entre colchetes `[]`, enquanto os parâmetros obrigatórios aparecem entre chaves `{}`. As opções disponíveis incluem as seguintes:

- h O elemento pode ser posicionado na mesma posição em que ele aparece no código fonte do texto.

$$ax^2 + bx + c = 0$$

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Tabela 4.1: Equação de segundo grau

Tabela 4.2: Raízes da equação da tabela 4.1

- t O elemento pode ser posicionado no topo de uma página.
- b O elemento pode ser posicionado no fim de uma página.
- p O elemento pode ser incluído em uma página formada só por flutuantes.
- ! Normalmente o L<sup>A</sup>T<sub>E</sub>X faz algumas considerações de ordem estética no posicionamento dos flutuantes, o que às vezes faz com que alguns elementos sejam posicionados muito longe de onde são citados, principalmente se você não incluir a opção p. Para fazer com que as considerações estéticas não sejam levadas em conta para um dado elemento, inclua a opção !.

## 4.2 Tabelas em L<sup>A</sup>T<sub>E</sub>X

Tabelas são construídas com comandos próprios do L<sup>A</sup>T<sub>E</sub>X, notadamente o ambiente `tabular`. Nada obriga a que o ambiente `tabular` esteja sempre posicionado em um elemento flutuante. Se você quiser impor que uma tabela fique obrigatoriamente em uma determinada posição do texto, basta não colocar o `tabular` dentro de um `table`. Tabelas podem até ser incluídas no meio de uma frase. Por exemplo, eu posso dizer que se um jogo da velha está na configuração 

x		x
x	o	o

 e se o jogador “x” sabe jogar, então o jogador “o” irá perder, independentemente da jogada que faça.


O ambiente `tabular` tem um parâmetro obrigatório que indica o número de colunas da tabela e o posicionamento dos objetos em cada coluna. Por exemplo, uma tabela criada com `\begin{tabular}{lcr}` terá três colunas; o texto será alinhado à esquerda na primeira coluna, centralizado na segunda e alinhado à direita na terceira. Podem ser incluídos objetos que ocupam mais de uma linha (comando `multirow`) ou mais de uma coluna (comando `multicolumn`). Neste último caso, também é possível mudar o alinhamento do texto. Exemplos podem ser vistos nas tabelas 3.1 e 3.3, na página 15.

Com o pacote `tabularx`, além das opções normais de posicionamento de colunas (`lcr`), pode-se incluir automaticamente um texto qualquer antes de cada elemento da coluna (`>{}`). Este recurso foi utilizado nas tabelas 3.1 e 3.3 para fazer com que todos os textos de algumas colunas fossem automaticamente escritos na fonte `tt`. Além disso, podem-se criar colunas de largura fixa e/ou de largura que se ajustam para que a tabela ocupe toda a largura desejada, além do estilo tradicional de coluna que assume a largura suficiente para conter seus elementos. Exemplos de colunas com diferentes larguras e alinhamentos podem ser vistos na tabela 4.3.

## 4.3 Figuras em L<sup>A</sup>T<sub>E</sub>X

As figuras (imagens, desenhos, gráficos, etc.) devem ser produzidas por ferramentas externas ao L<sup>A</sup>T<sub>E</sub>X, salvas em um arquivo e inseridas no texto usando o comando `includegraphics`. Da mesma forma que as tabelas, as figuras podem ser flutuantes, caso sejam inseridas dentro de um ambiente `figure`, ou ter uma posição fixa no texto



(como aqui: ).

COLUNA p	COLUNA X	COLUNA l
Largura fixa (não depende do conteúdo)	Expandível	Ajustável
Alinhada no topo	Alinhada à esquerda	Alinhada à esquerda

COLUNA b	COLUNA C (ver comandos.tex)	COLUNA r
Largura fixa (não depende do conteúdo)	Expandível	Ajustável
Alinhada na base	Centralizada	Alinhada à direita

Tabela 4.3: Tabelas com colunas de diferentes larguras e alinhamentos

O formato em que você deve salvar os arquivos das figuras para que possa incluí-las no texto depende de como você pretende compilar o código fonte:

- se o texto vai ser compilado com `latex`, todos os arquivos devem estar no formato EPS (*Encapsulated PostScript*);
- se o texto vai ser compilado com `pdflatex`, os arquivos devem estar nos formatos PDF ou JPEG (outros formatos são aceitos, mas estes são os recomendáveis).

É aconselhável que você não inclua a terminação no nome do arquivo que é parâmetro para o comando `includegraphics`. Isto porque, de acordo com a forma como o texto está sendo compilado, o  $\text{\LaTeX}$  acrescenta a terminação adequada. Por exemplo, caso seu texto inclua o comando `\includegraphics{eu}`, o  $\text{\LaTeX}$  procurará o arquivo `eu.eps` caso esteja sendo chamado via `latex` ou um dos arquivos `eu.pdf` ou `eu.jpg` caso esteja sendo chamado via `pdflatex`.

As figuras podem ser divididas em dois grandes grupos:

- As imagens e fotos, que normalmente correspondem a visões reais do mundo e são obtidas por câmeras digitais ou assemelhados. Caracterizam-se por conterem grandes quantidades de nuances, texturas e cores.
- As figuras sintéticas, normalmente produzidas utilizando *softwares* dedicados. Geralmente contêm figuras geométricas (linhas, quadrados, etc.), textos e poucas cores e texturas. Neste grupo, para efeito de discussão das ferramentas de produção, podem-se identificar duas categorias:
  - Os desenhos e esquemas: diagramas de blocos, organogramas e fluxogramas, representações esquemáticas, etc.
  - Os gráficos: representações gráficas de valores ou funções matemáticas.

### 4.3.1 Imagens e fotos

As imagens e fotos normalmente só podem ser armazenadas em formatos que representam cada *pixel* da imagem separadamente, eventualmente com algum tipo de compressão. Os formatos JPEG, GIF, TIF, PNM (PBM, PGM ou PPM), BMP (Bitmap) e PNG,

entre outros, são todos desta categoria. Se sua figura está em algum destes formatos, você deve convertê-la para EPS (se usar `latex`) ou para JPEG (se usar `pdflatex`) para poder incluí-la no documento  $\text{\LaTeX}$ .

A quase totalidade dos *softwares* de visualização de imagens permite salvá-las em múltiplos formatos, geralmente incluindo JPEG e EPS. No Unix, você dispõe ainda de vários programas para fazer a conversão em comandos de linha: `jpegtopnm`, `pnmtojpeg`, `pnmtops`, `gif2ps`, `giftopnm`, `tiff2ps`, `tifftopnm`, `bmptopnm` e `pngtopnm`, entre outros.

A figura 4.2 mostra um exemplo de inclusão de uma imagem no texto  $\text{\LaTeX}$ .



Figura 4.2: Exemplo de imagem real

### 4.3.2 Figuras sintéticas

As figuras sintéticas podem ser armazenadas em formato *pixel-a-pixel*, como se fossem uma imagem, ou em formato vetorial. No formato vetorial as primitivas que formam a figura (linhas, textos, etc.) são descritas pelos parâmetros que as caracterizam (ponto de início e fim, *string* e posição do texto, etc.). As figuras em formato vetorial são mais adequadas pois usualmente correspondem a arquivos menores e a qualidade da imagem não sofre perdas ao se aumentar ou diminuir o tamanho da figura.

Para inclusão no  $\text{\LaTeX}$ , os formatos PDF e EPS são os únicos que podem representar figuras no formato vetorial. Nem toda figura salva nestes formatos, entretanto, é necessariamente vetorial, pois tanto o PDF quanto o EPS podem representar tanto figuras em formato *pixel-a-pixel* quanto figuras em formato vetorial. Para que sua figura seja vetorial, é necessário que o *software* que a gerou tenha a capacidade de produzi-las.

Para demonstrar a melhor qualidade das figuras em formato vetorial, nas figuras 4.3 e 4.4 se mostra em tamanho natural um mesmo diagrama nos formatos vetorial e de *pixels*. Nas figuras 4.5 e 4.6 estas mesmas figuras são apresentadas com uma redução de 50%, utilizando o parâmetro `scale` do `includegraphics`. Já nas figuras 4.7 e 4.8 o diagrama original foi reduzido, de forma que seu tamanho natural é menor. Nas figuras

4.9 e 4.10 este diagrama pequeno está aumentado de um fator arbitrário, calculado pelo `includegraphics` para que a imagem ocupe toda a largura da linha.

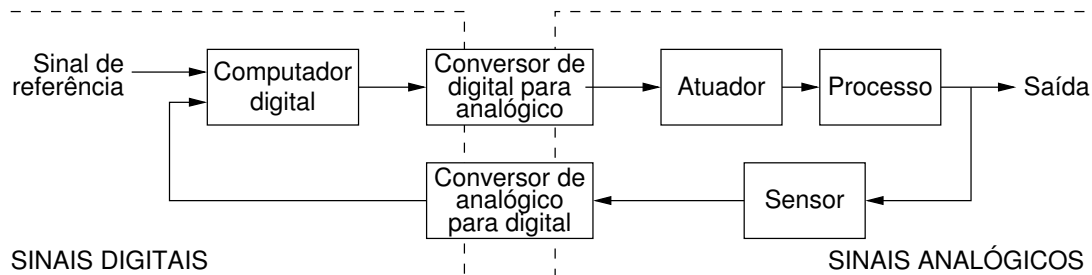


Figura 4.3: Figura vetorial grande em tamanho natural

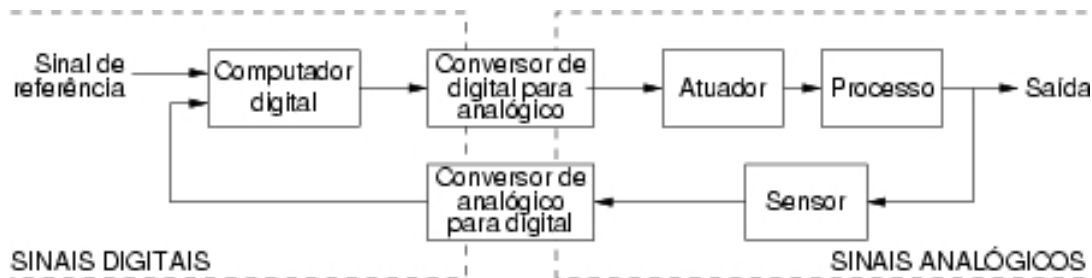


Figura 4.4: Figura *pixel-a-pixel* grande em tamanho natural

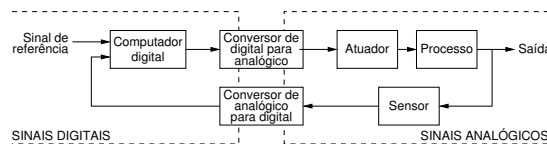


Figura 4.5: Figura vetorial grande em tamanho reduzido

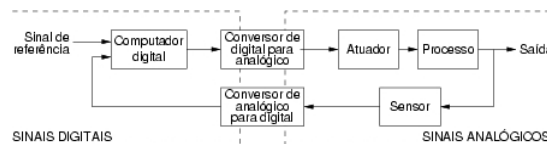


Figura 4.6: Figura *pixel-a-pixel* grande em tamanho reduzido

Nota-se que no formato vetorial as linhas mantêm a espessura mesmo quando se fazem ampliações ou reduções. Já no formato de *pixels* as linhas ficam mais claras (cinzas, ao invés de pretas) após as reduções e mais grossas após as ampliações, além de uma perda geral de definição da imagem.



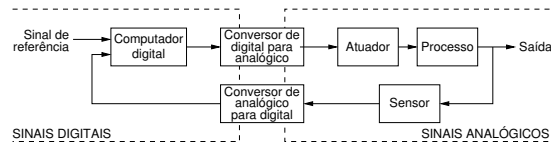


Figura 4.7: Figura vetorial pequena em tamanho natural

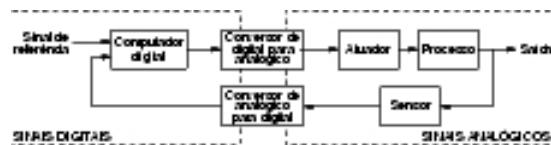
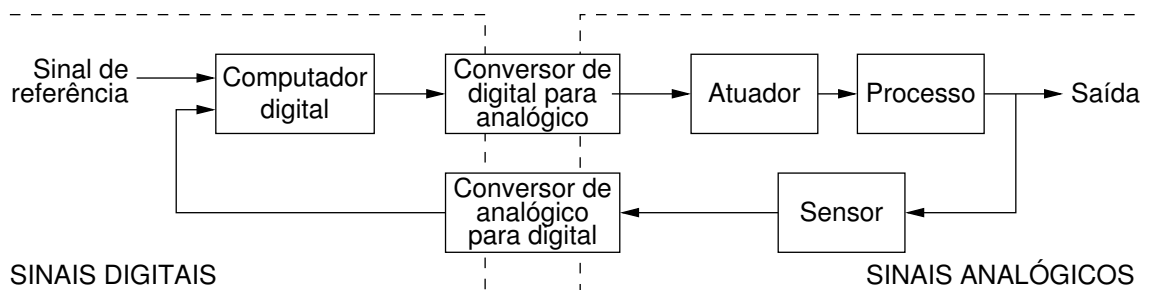
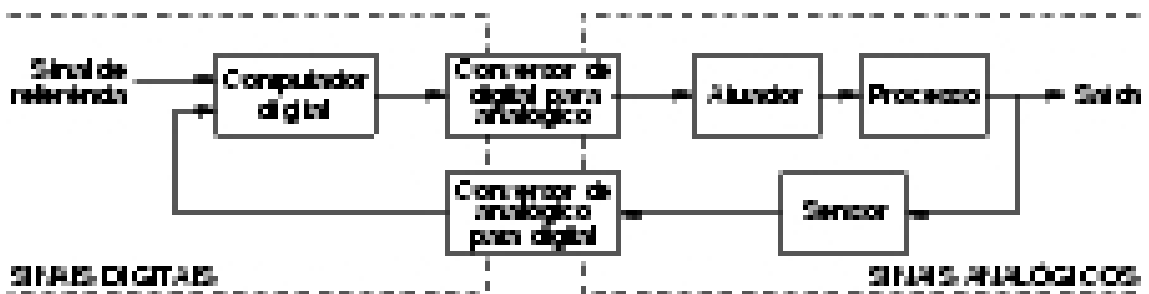
Figura 4.8: Figura *pixel-a-pixel* pequena em tamanho natural

Figura 4.9: Figura vetorial pequena em tamanho ampliado

Figura 4.10: Figura *pixel-a-pixel* pequena em tamanho ampliado

## 4.4 Ferramentas para desenhos e esquemas

Existem diversas ferramentas para fazer desenhos, mas muitas delas apenas salvam a figura gerada em formatos *pixel-a-pixel*. No Unix, pode-se utilizar o `xfig`, que exporta imagens em muitos formatos, inclusive nos vetoriais (PDF e EPS). Os diagramas das figuras 4.3 a 4.10 foram desenhados e exportados no `xfig`. O arquivo fonte correspondente é o `diagrama.fig`, no diretório `figuras`.

A possibilidade de salvar figuras em modo vetorial impõe que alguns recursos para desenho de imagens não sejam oferecidos. Um deles é o desenho a mão-livre, já que seria impossível descrever a curva obtida em termos de figuras geométricas básicas. Outro recurso inexistente é o de preencher uma região com uma determinada cor. Esta última limitação muitas vezes pode ser contornada utilizando-se a noção de profundidade. Por exemplo, para desenhar uma figura vazado e preenchido de azul, pode-se desenhar a figura externa preenchido de azul sobre o qual se desenha a figura interna preenchido de branco, como mostram os exemplos da figura 4.11.

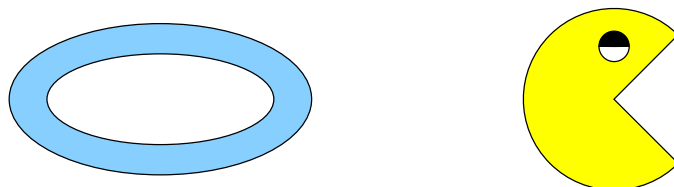


Figura 4.11: Preenchimento de figuras utilizando diferentes profundidades

A noção de profundidade no `xfig` foi exaustivamente utilizada para desenhar os símbolos da UFRN e do PPgEEC que podem ser vistos na página de rosto deste documento. Os arquivos `xfig` correspondentes são `UFRN.fig` e `PPgEE.fig`. Ela também pode ser utilizada para mesclar imagens com figuras sintéticas, como na figura 4.12 (veja arquivo `figuras/pensador.fig`).



Figura 4.12: Imagem mesclada com elementos sintéticos

Outra possibilidade oferecida pelo `xfig` é a inclusão de comandos  $\text{\LaTeX}$  dentro da figura. Para utilizar este recurso, marque no `xfig` os textos que devem ser interpretados como comandos  $\text{\LaTeX}$  com o *flag* `special` e exporte a figura no modo *Combinado PS/Latex* ou *Combinado PDF/Latex*. Veja um exemplo na figura 4.13; note que o arquivo é incluído com `\input{}` e não com `\includegraphics{}`.

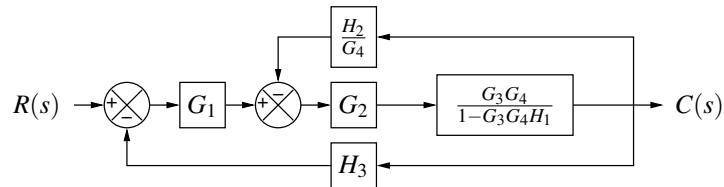


Figura 4.13: Figura incluindo comandos  $\text{\LaTeX}$

## 4.5 Ferramentas para gráficos

Gráficos devem ser gerados com aplicativos capazes de exportar o resultado nos formatos EPS ou PDF, preferencialmente em formato vetorial. Os conhecidos programas *Scilab* e *Matlab* têm esta capacidade. Se você deseja algo mais simples, a ferramenta *GNUplot* é uma das mais utilizadas no Unix para a geração de gráficos de funções matemáticas.

Uma vez gerados, gráficos são inseridos no texto tal como figuras. A figura 4.14 apresenta um gráfico gerado através do comando de linha `gnuplot grafico.gnuplot`. Este arquivo `grafico.gnuplot`, que contém uma série de comandos do *GNUplot*, está no diretório `figuras`.

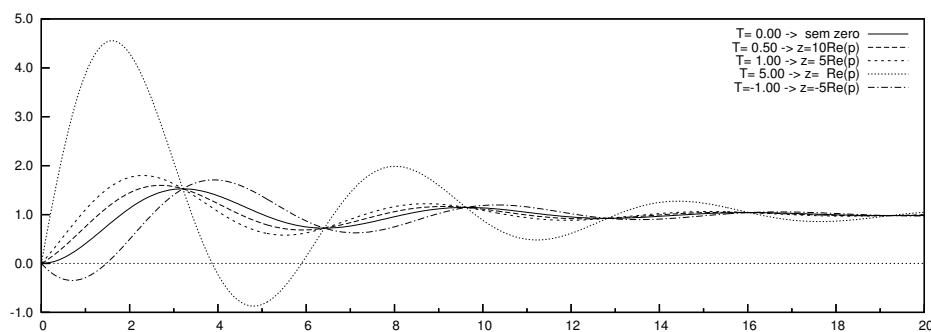


Figura 4.14: Exemplo de gráfico de funções matemáticas

## 4.6 Conclusões

Ferramentas de desenho capazes de gerar a saída em formato vetorial são mais difíceis de usar e parecem ser dotadas de menos recursos do que outras que só exportam seus

resultados como imagens de *pixels*. Isto se deve à necessidade de descrever todos os elementos da imagem sob a forma de primitivas parametrizáveis para permitir que elas sejam escaláveis à vontade e exportáveis para qualquer formato desejado.

Entretanto, a qualidade visual das figuras obtidas e a sua reusabilidade é muito maior. A comparação é aproximadamente a mesma que a entre textos produzidos em  $\text{\LaTeX}$  e em editores gráficos. Desta forma, na medida do possível, tente conjugar a escrita do documento  $\text{\LaTeX}$  com a utilização de alguma ferramenta de desenho vetorial.

---

# Capítulo 5

## Implementação

---

Coloque aqui a parte sistêmica da solução do problema. Para isto, monte um diagrama de blocos, descrevendo cada caixinha do diagrama. Caso o seu trabalho lide com engenharia de *software*, você também pode apresentar também outros tipos de diagrama (casos de uso, classes, modelos entidade-relacionamento, entre outros), desde que saiba o que eles significam. Diga qual a linguagem, máquina e outras informações técnicas. Coloque os algoritmos ou técnicas implementadas que resolveram o problema especificado formalmente no capítulo anterior. Use e abuse de textos como “este algoritmo implementa a equação xx do Capítulo YY”, para fazer referência aos formalismos definidos nos capítulos 2 e 4.

### 5.1 Algoritmos

O latex provê alguns pacotes para a montagem de algoritmos. Um destes pacotes é o `algorithm2e`, que permite o uso de diretivas de comando em português. De acordo com o manual do pacote, os comandos disponíveis em português são os seguintes:

- Entradas e saídas do algoritmo:
  - `\Entrada{Entrada}` → **KwEntrada**;
  - `\Saida{Saída}` → **KwSaida**;
  - `\Dados{Dados}` → **KwDados**;
  - `\Resultado{Resultado}` → **KwResultado**.
- Retorno do algoritmo:
  - `\Ate` → **até**;
  - `\KwRetorna{[valor]}` → **KwRetorna**;
  - `\Retorna{[valor]}` → **Retorna**.
- Início de bloco: `\Iniciob{bloco interno}` → **Iniciob**.
- Condicionais (*if-then-else*):
  - `\eSe{condição}{bloco 'then'}{bloco 'else'}` → **eSe**;
  - `\Se{condição}{bloco 'then'}` → **Se**;
  - `\uSe{condição}{bloco 'else'}` sem 'fim' → **uSe**;
  - `\lSe{condição}{linha de texto 'then'}` → **lSe**;

- \Senao{bloco 'else'} → **Senão**;
  - \uSenao{bloco 'else' sem 'else'} → **uSenão**;
  - \lSenao{linha de código ``else''} → **lSenão**;
  - \SenaoSe{condição}{bloco ``else-if''} → **uSenãoSe**;
  - \uSenaoSe{condição}{bloco ``else-if'' sem ``fim''} → **uSenãoSe**;
  - \lSenaoSe{condição}{linha de código ``else-if''} → **lSenãoSe**.
- Seleção de casos (*switch-case*):
    - \Selec{condição}{bloco 'switch'} → **Selec**;
    - \Caso{um caso}{bloco 'case'} → **Caso**;
    - \uCaso{um caso}{bloco 'case' sem 'fim'} → **uCaso**;
    - \lCaso{um caso}{linha de código 'case'} → **lCaso**;
    - \Outro{bloco 'outros'} → **Outro**;
    - \lOutro{linha de código 'outros'} → **lOutro**.
  - Laços *for*:
    - \Para{condição}{bloco de repetição} → **Para**;
    - \lPara{condição}{linha de código de repetição} → **lPara**;
    - \ParaPar{condição}{bloco de repetição} → **ParaPar**;
    - \lParaPar{condição}{linha de código de repetição} → **lParaPar** ;
    - \ParaCada{condição}{bloco de repetição} → **ParaCada**;
    - \lParaCada{condição}{linha de código de repetição} → **lParaCada**;
    - \ParaTodo{condição}{bloco de repetição} → **ParaTodo**;
    - \lParaTodo{condição}{linha de código de repetição} → **lParaTodo**.
  - Laços *while*:
    - \Enqto{condição de parada}{bloco de repetição} → **Enqto**;
    - \lEnqto{condição de parada}{bloco de repetição} → **lEnqto**.
  - Laços *do-while*:
    - \Repita{condição de parada}{bloco de repetição} → **Repita**;
    - \lRepita{condição de parada}{linha de código de repetição} → **lRepita**.
  - Comentários: \tcc{comentário}.

Os algoritmos 1 e 2 apresentam alguns modelos de algoritmos para serem tomados de exemplo.

---

**Algoritmo 1:** Algoritmo para interpolação de Lagrange.

---

**Entrada:**  $x$ : vetores de valores;  $y = L(x)$ ;  $p$ : valor de entrada a ser calculado**Saída:**  $s = L(p)$ 

```
1  $n \leftarrow \text{comprimento}(x)$ ;  
2  $s \leftarrow 0$ ;  
3 para  $i = 1$  até  $n$  faça  
4    $L \leftarrow 1$ ;  
5   para  $j = 1 : 1 : n$  faça  
6     se  $i \neq j$  então  
7        $L \leftarrow L * \left( \frac{p - x[j]}{x[i] - x[j]} \right)$   
8     fim  
9   fim  
10   $s \leftarrow s + L * y[i]$ ;  
11 fim  
12 retorna  $s$ ;
```

---

---

**Algoritmo 2:** Algoritmo para a integração pelo primeiro método de Simpson.**Entrada:**  $a$ : valor inicial;  $b$ : valor final;  $n$ : número de subintervalos (deve ser múltiplo de 2)/\* A função a ser integrada é definida em uma função denominada  $f$ , fora do escopo deste algoritmo. \*/**Saída:**  $I$  = integral de  $f$  entre  $a$  e  $b$ 

```
1  $h \leftarrow \frac{b-a}{n}$ ;
2  $x[1] \leftarrow a$ ;
3  $y[1] \leftarrow f(a)$ ;
4  $I \leftarrow 0$ ;
5  $k \leftarrow 2$ ;
6 enquanto  $k \leq n$  faça
7    $x[i] \leftarrow x[i-1] + h$ ;
8    $y[i] \leftarrow f(x[i])$ ;
9   se  $i \% 2 = 0$  então
10     $I \leftarrow I + 4 * y[i]$ ;
11  senão
12     $I \leftarrow I + 2 * y[i]$ ;
13  fim
14   $k = k + 1$ ;
15 fim
16  $x[n+1] \leftarrow b$ ;
17  $y[n+1] \leftarrow f(x[n+1])$ ;
18  $I \leftarrow I + \frac{h}{3} * (I + y[n+1])$ ;
19 retorna  $I$ ;
```

---



---

## Capítulo 6

# Experimentos e Resultados

---

Esta é a parte mais importante (e interessante) do trabalho. Especifique o ferramental usado para a experimentação empírica da solução implementada. Descreva cada experimento realizado de forma bem explicativa (ou seja, planeje os experimentos). Veja o que experimentar, o que são resultados (erros e métricas usadas para medi-los, medidas estatísticas). Use e abuse de gráficos, tabelas e figuras mostrando resultados visuais (impressiona). Uma dica interessante: faça um video e coloque no youtube, demonstrando o sistema funcionando (impressiona), citando a URL entre parênteses. Não apenas descreva os experimentos, mas também discuta os resultados e os compare (principalmente) com outros encontrados na literatura (e citados nos trabalhos relacionados).



---

# Capítulo 7

## Conclusão

---

O capítulo final depende do tipo de documento. Nas propostas de tema deve ser apresentado de forma clara e sucinta o assunto a ser desenvolvido e o cronograma de execução do trabalho. Nas teses e dissertações devem ser ressaltadas as principais contribuições do trabalho e as suas limitações.

As contribuições devem evitar as adjetivações e julgamentos de valor. Quanto às limitações, não tenha medo de as apresentar: é muito mais reconhecido um autor que apresenta os casos em que sua proposta não se aplica do que outro que parece não ter consciência deles.

Escreva (com outras palavras) o que foi realizado e como foi realizado, o que o trabalho descrito no artigo conseguiu melhorar e qual a sua relevância, e quais são as vantagens e limitações das propostas que a tese/dissertação apresenta. Apresente também eventuais aplicações dos resultados obtidos (ou da metodologia, técnica, produto) e ideias de trabalhos futuro que possam melhorar o seu (não apenas apresente, mas indique como pode ser feito).

### 7.1 Etapas de Homologação do Título

Depois de defendido, o seu trabalho passará por um processo de homologação. Atualmente, este pode ser feito e acompanhado via SiGAA, menu Ensino → Produções Acadêmicas → Acompanhar Procedimentos após Defesa.

1. A consolidação da atividade de defesa;
2. A submissão da versão final corrigida da dissertação. Nesta etapa, o seu orientador irá verificar se as falhas apontadas pela banca foram corrigidas. Caso isto tenha acontecido, a sua dissertação/tese será considerada aprovada, passando para a próxima etapa. Caso contrário, você terá que submeter uma nova versão final, para corrigir os erros apontados pelo seu orientador;
3. A submissão da versão final com a ficha catalográfica, cujas informações podem ser obtidas na biblioteca central, ou solicitadas pelo SiGAA, no *link* “Solicitar Ficha Catalográfica”. Esta também será avaliada pelo seu orientador. Caso seja aprovada, virá o próximo passo;
4. A assinatura do termo de autorização da publicação, que pode ser feita via SiGAA, menu Ensino → Termo de Autorização;

5. O envio da versão final para a avaliação da coordenação. Neste ponto será avaliado se o sua monografia/tese satisfaz os requisitos estabelecidos pelo programa. Aqui, **é muito importante que todas as considerações fornecidas neste modelo seja seguidas rigorosamente**, caso contrário, o seu trabalho será reenviado para fazer as correções necessárias. Caso esteja conforme o exigido pela coordenação, este será aprovado;
6. A solicitação de homologação do diploma, que pode ser acompanhada na Pro-Reitoria de Pós-Graduação.

## 7.2 Para saber mais

Procure no Google, ora! Brincadeiras a parte, existem inúmeros tutoriais sobre  $\text{\LaTeX}$  na rede que podem dar maiores informações sobre o aplicativo. Para conhecer os pacotes disponíveis, uma opção é o livro *The  $\text{\LaTeX}$  Companion* (?), popularmente conhecido como o “livro do cachorro”. Outras informações sobre redação técnica e normas para confecção de teses e dissertações podem ser encontradas em livros de Metodologia Científica.

---

# Apêndice A

## Informações adicionais

---

Os apêndices são normalmente empregados para incluir informações adicionais a serem eventualmente consultadas mas que não são essenciais para a compreensão do texto.

Evite sobrecarregar seu texto com informações longas e de pouco interesse para uma primeira leitura. São normalmente colocados nos apêndices:

- longas deduções ou demonstrações de fórmulas e teoremas;
- especificações técnicas de equipamentos e descrições de experimentos;
- eventuais conhecimentos disponíveis na literatura mas que se julga conveniente repetir no texto para facilitar a compreensão do leitor não familiarizado com a área;
- outras informações que se julga que devam ser preservadas mas que não são importantes no documento, tais como diagramas esquemáticos, algoritmos ou trechos de código-fonte, folhas de especificações, etc.