

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

ANA LETÍCIA HERCULANO DA SILVA

**Técnicas e Critérios de Testes em uma
Aplicação de Banco de Dados
Relacioanal
Estudo de Caso**

Goiânia
2017

UNIVERSIDADE FEDERAL DE GOIÁS
INSTITUTO DE INFORMÁTICA

**AUTORIZAÇÃO PARA PUBLICAÇÃO DE TRABALHO DE
CONCLUSÃO DE CURSO EM FORMATO ELETRÔNICO**

Na qualidade de titular dos direitos de autor, **AUTORIZO** o Instituto de Informática da Universidade Federal de Goiás – UFG a reproduzir, inclusive em outro formato ou mídia e através de armazenamento permanente ou temporário, bem como a publicar na rede mundial de computadores (*Internet*) e na biblioteca virtual da UFG, entendendo-se os termos “reproduzir” e “publicar” conforme definições dos incisos VI e I, respectivamente, do artigo 5º da Lei nº 9610/98 de 10/02/1998, a obra abaixo especificada, sem que me seja devido pagamento a título de direitos autorais, desde que a reprodução e/ou publicação tenham a finalidade exclusiva de uso por quem a consulta, e a título de divulgação da produção acadêmica gerada pela Universidade, a partir desta data.

Título: Técnicas e Critérios de Testes em uma Aplicação de Banco de Dados Relacional – Estudo de Caso

Autor(a): Ana Letícia Herculano da Silva

Goiânia, 20 de Julho de 2017.

Ana Letícia Herculano da Silva – Autor

Cássio Leonardo Rodrigues – Orientador

Dr. Edmundo Sérgio Spoto – Co-Orientador

ANA LETÍCIA HERCULANO DA SILVA

Técnicas e Critérios de Testes em uma Aplicação de Banco de Dados Relacionai

Estudo de Caso

Trabalho de Conclusão apresentado à Coordenação do Curso de Sistemas de Informação do Instituto de Informática da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Área de concentração: Teste de Software.

Orientador: Prof. Cássio Leonardo Rodrigues

Co-Orientador: Prof. Dr. Edmundo Sérgio Spoto

Goiânia
2017

ANA LETÍCIA HERCULANO DA SILVA

Técnicas e Critérios de Testes em uma Aplicação de Banco de Dados Relacionais

Estudo de Caso

Trabalho de Conclusão apresentado à Coordenação do Curso de Sistemas de Informação do Instituto de Informática da Universidade Federal de Goiás como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação, aprovada em 20 de Julho de 2017, pela Banca Examinadora constituída pelos professores:

Prof. Cássio Leonardo Rodrigues

Instituto de Informática – UFG

Presidente da Banca

Prof. Dr. Edmundo Sérgio Spoto

Instituto de Informática – UFG

Prof. Nome do membro da banca

Unidade acadêmica – Sigla da universidade

Profa. Nome do membro da banca

Unidade acadêmica – Sigla da universidade

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador(a).

Ana Letícia Herculano da Silva

<Texto com um perfil resumido do autor do trabalho. Por exemplo: (Graduou-se em Artes Cênicas na UFG - Universidade Federal de Goiás. Durante sua graduação, foi monitor no departamento de Filosofia da UFG e pesquisador do CNPq em um trabalho de iniciação científica no departamento de Biologia. Durante o Mestrado, na USP - Universidade de São Paulo, foi bolsista da FAPESP e desenvolveu um trabalho teórico na resolução do Problema das Torres de Hanói. Atualmente desenvolve soluções para problemas de balanceamento de ração para a pecuária de corte.)>

<Dedicatória do trabalho a alguma pessoa, entidade, etc.>

Agradecimentos

<Texto com agradecimentos àquelas pessoas/entidades que, na opinião do autor, deram alguma contribuição relevante para o desenvolvimento do trabalho.>

<Epígrafe é uma citação relacionada com o tópico do texto>

**<Nome do autor da citação>,
<Título da referência à qual a citação pertence>.**

Resumo

Silva, Ana Letícia Herculano da. **Técnicas e Critérios de Testes em uma Aplicação de Banco de Dados Relacioanal**. Goiânia, 2017. 52p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

A garantia de qualidade de um software é importante para agregação de valor aos clientes que o utilizam. Uma das fases para a obtenção dessa qualidade é a de Testes de Software. O testes de software pode ser aplicado em diferentes níveis e de diferentes técnicas. A proposta para aumentar a confiabilidade do software é aplicar técnicas de testes funcional ou baseado em erros ou estrutural com base nas dependência de dados e integridade das informações trabalhadas no Banco de Dados. Muitas vezes todas as técnicas podem ser aplicadas de forma complementar, já que nenhuma inclui a outra. Neste Trabalho foi aplicado um estudo de caso de um software real, onde é aplicado técnicas e critérios de testes funcional e teste de mutantes em uma Aplicação de Banco de Dados Relacional. São apresentados a contextualização teórica, um estudo de caso, os resultados obtidos e uma análise geral desses resultados.

Palavras-chave

Teste de Software, Banco de Dados, Teste Funcional e Teste de Mutantes

Abstract

Silva, Ana Letícia Herculano da. <Work title>. Goiânia, 2017. 52p. Relatório de Graduação. Instituto de Informática, Universidade Federal de Goiás.

The book is on the table.

Keywords

Software Testing, Database, Fuctional Testing, Mutant Testing

Sumário

Lista de Figuras	11
Lista de Tabelas	12
Lista de Algoritmos	13
Lista de Códigos de Programas	14
1 Introdução	15
2 Descrição da classe inf-ufg	16
2.1 Opções da classe	16
2.2 Parâmetros da classe	16
2.3 Elementos Pré-Textuais	17
3 Elementos do texto	20
3.1 Figuras	20
3.1.1 Subfiguras	22
3.2 Tabelas	23
3.3 Algoritmos	23
3.4 Códigos de Programa	24
3.5 Teoremas, Corolários e Demonstrações	25
3.6 Citações Longas	26
3.7 Referências Bibliográficas	26
4 Introdução	29
5 Descrição da classe inf-ufg	31
5.1 Opções da classe	31
5.2 Parâmetros da classe	31
5.3 Elementos Pré-Textuais	32
6 Elementos do texto	35
6.1 Figuras	35
6.1.1 Subfiguras	37
6.2 Tabelas	38
6.3 Algoritmos	38
6.4 Códigos de Programa	39
6.5 Teoremas, Corolários e Demonstrações	40

6.6	Citações Longas	41
6.7	Referências Bibliográficas	41
	Referências Bibliográficas	44
A	Exemplo de um Apêndice	45
B	Exemplo de Outro Apêndice	49

Lista de Figuras

3.1	Uma figura típica.	21
3.2	Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.	21
3.3	Figura incluída no texto com a classe <code>graphicx</code> .	22
3.4	(a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.	22
	(b) Segunda subfigura (um pedaço).	22
6.1	Uma figura típica.	36
6.2	Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.	36
6.3	Figura incluída no texto com a classe <code>graphicx</code> .	37
6.4	(a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.	37
	(b) Segunda subfigura (um pedaço).	37

Lista de Tabelas

3.1	Conteúdo do diretório [?]	28
6.1	Conteúdo do diretório [?]	43

Lista de Algoritmos

3.1	$MSR(A, i, j)$	24
6.1	$MSR(A, i, j)$	39

Lista de Códigos de Programas

3.1	<code>insertionsort()</code>	25
6.1	<code>insertionsort()</code>	40

Introdução

A onipresença de software ao redor do globo é indiscutível. Consumimos (e produzimos) em uma escala imensurável pelos mais otimistas de tempos passados. Temos contato

A qualidade de software aos poucos vem ocupando seu espaço no processo de desenvolvimento de um software. O testes de software em si, é a principal etapa para que a qualidade seja garantida. Foram realizados alguns trabalhos e pesquisa com base nas diversas técnicas de testes, importantes para a realização deste trabalho em si.

Souza [7] realizou uma pesquisa a qual foi a base para alguns critérios de testes aplicados neste trabalho. Ela propôs alguns critérios de testes em Banco de Dados Relacional baseados na especificação de requisitos através da UML. Para apoiar a aplicação dos conjuntos de critérios propostos, foi desenvolvido uma abordagem denominada mapeamento Conceitual de Informação.

Descrição da classe inf-ufg

2.1 Opções da classe

Para usar esta classe num documento $\text{\LaTeX} 2_{\epsilon}$, coloque os arquivos `inf-ufg.cls`, `inf-ufg.bst`, `abnt-num.bst`, `atbeginend.sty` e `tocloft.sty` numa pasta onde o compilador \LaTeX pode achá-lo (normalmente na mesma pasta que seu arquivo `.tex`), e defina-o como o estilo do seu documento. Por exemplo, uma dissertação de mestrado que usa o modelo abnt de citações bibliográficas:

```
\documentclass[dissertacao,abnt]{inf-ufg}
...
\begin{document}
```

As opções da classe são `[tese]` (para tese de doutorado), `[dissertacao]` (para dissertação de mestrado), `[monografia]` (para monografia de curso de especialização e `[relatorio]` (para relatório final de curso de graduação). Se nenhuma opção for declarada, o documento é considerado como uma dissertação de mestrado. Se a opção `[abnt]` for utilizada, as citações bibliográficas serão geradas conforme definido pelo grupo de trabalho `abnt-tex`. Contudo, o mais recomendável é não utilizar essa opção. Com a opção `[nocolorlinks]` todos os *links* de navegação no texto ficam na cor preta. O ideal é usar esta opção para gerar o arquivo para impressão, pois a qualidade da impressão dos *links* fica superior.

2.2 Parâmetros da classe

Os elementos pré-textuais são definidos página por página e dependem da correta definição dos parâmetros listados a seguir (aqueles que contém um texto/valor padrão não precisam ser definidos, caso atenda a situação do autor do texto que está usando a classe `inf-ufg.cls`):

- `\autor` : Nome completo do autor da tese, começando pelo apelido (ex.: José da Silva);

- `\autorR` : Nome completo do autor da tese, começando pelo nome (ex.: da Silva, José);
- `\titulo` : Título da tese, dissertação, monografia ou relatório de conclusão de curso;
- `\subtitulo` : Se tiver um subtítulo, use este macro para defini-lo;
- `\cidade` : A cidade de edição. A cidade padrão é Goiânia.
- `\dia` : Dia do mês da data de defesa (1–31);
- `\mes` : Mês da data de defesa (1–12);
- `\ano` : Ano da data de defesa;
- `\universidade` : Nome completo da universidade. O nome padrão é Universidade Federal de Goiás;
- `\uni` : Sigla da universidade. A sigla padrão é UFG;
- `\unidade` : Nome da unidade acadêmica. O padrão é Instituto de Informática;
- `\departamento` : Nome do departamento, com maiúscula na primeira letra (para o caso de unidades com mais de um departamento);
- `\programa` : Nome do programa de pós-graduação, com maiúscula na primeira letra. O padrão é Computação;
- `\concentracao` : Nome da área de concentração;
- `\orientador` : Nome completo do orientador, começando pelo apelido;
- `\orientadorR` : Nome completo do orientador, começando pelo nome;
- `\orientadora` : Nome completo da orientadora, começando pelo apelido; use este comando e o próximo se for orientadora e não orientador.
- `\orientadoraR` : Nome completo do orientadora, começando pelo nome;
- `\coorientador` : Nome completo do co-orientador, começando pelo apelido;
- `\coorientadorR` : Nome completo do co-orientador, começando pelo nome;
- `\coorientadora` : Nome completo da coorientadora, começando pelo apelido; use este comando e o próximo se for coorientadora e não coorientador.
- `\coorientadoraR` : Nome completo do coorientadora, começando pelo nome;
- `\universidadeco` : Nome da universidade do coorientador;
- `\unico` : Sigla da universidade do coorientador;
- `\unidadeco` : Nome da unidade acadêmica do coorientador.¹

2.3 Elementos Pré-Textuais

Os elementos pré-textuais são definidos página por página, conforme descritos a seguir:

¹ Se não tiver um co-orientador, não defina esses últimos sete parâmetros.

capa

`\capa` : Gera o modelo da capa externa do trabalho. Esta página servirá apenas como modelo para a encadernação da versão final do texto. Nenhum dado é necessário.

publicação

`\publica` : Gera a autorização para publicação do trabalho em formato eletrônico e disponibilização do mesmo na biblioteca virtual da UFG.

rosto

`\rosto` : Gera a folha de rosto, a qual é a primeira folha interna do trabalho. Nenhum dado é necessário.

aprovação

`\aprovacao` : ambiente para a reprodução do termo de aprovação da Banca Examinadora da tese ou dissertação.

banca

`\banca` : Entrada para o nome dos examinadores, exceto o(s) orientador(es).

`\profa` : Entrada para o nome das examinadoras, exceto o(s) orientador(es).

direitos

`\direitos` : Macro com 2 argumentos para gerar os direitos autorais, o perfil do aluno e a ficha catalográfica da Biblioteca Central da UFG.

- O primeiro argumento é o Perfil do aluno; e
- O segundo argumento é a lista das palavras-chaves para a Ficha Catalográfica.

dedicatória

`\dedicatoria` : ambiente para escrever a dedicatória. É possível trocar o espaçamento dentro desse ambiente do mesmo jeito que no \LaTeX padrão.

agradecimentos

`\agradecimentos` : ambiente para escrever os agradecimentos. É possível trocar o espaçamento dentro desse ambiente do mesmo jeito que no \LaTeX padrão.

resumo

`\chaves` : A lista das palavras chaves, separadas por ‘;’. Deve ser definido antes do ambiente `\resumo`, o qual é usado para escrever o resumo em português.

abstract

`\keys` : A lista das palavras chaves em inglês, separadas por ‘;’. Deve ser definido antes do ambiente `\abstract`, o qual contém 1 argumento e é usado escrever o resumo em inglês. O argumento deve ser o título do trabalho em inglês.

tabelas

`\tabelas` : Macro com 1 argumento opcional para gerar as tabelas. O argumento pode ser:

- nada [] : gera apenas o sumário;
- fig : gera o sumário e uma lista de figuras;
- tab : gera o sumário e uma lista de tabelas;
- alg : gera o sumário e uma lista de algoritmos;
- cod : gera o sumário e uma lista de programas.

Pode-se usar qualquer combinação dessas opções. Por exemplo:

- figtab : gera o sumário e listas de figuras e tabelas,
- figtabcod : gera o sumário e listas de figuras, tabelas e códigos de programas;
- figtabalg : gera o sumário e listas de figuras, tabelas e algoritmos;
- figtabalgcod : gera o sumário e listas de figuras, tabelas, algoritmos e códigos de programas

epígrafe

`\epigrafe` : Macro com 3 argumentos que permite editar um epígrafe. O primeiro argumento é o texto da citação. O segundo argumento é o nome do autor da citação. O terceiro argumento é o título da referência à qual a citação pertence.

Elementos do texto

3.1 Figuras

Rótulos de figuras e tabelas devem ser centralizados se tiverem até uma linha (Figura 6.1), caso contrário devem estar justificados e identados em ambas as margens, como mostrado na Figura 6.2. Essa formatação já é realizada automaticamente pela classe `inf-ufg`.

Os compiladores \LaTeX provêem um mecanismo bastante simples para inclusão de figuras, o que pode ser feito com o auxílio de várias classes auxiliares (as mais comuns são `graphic` e `graphicx`). A classe `inf-ufg` usa o comando `\includegraphics`, da classe `graphicx`, para a inclusão de figuras e não é necessário você colocar a extensão do arquivo neste comando. Por exemplo, para a figura 6.1 os comandos usados foram:

```
\begin{figure}[htb]
\centering
\includegraphics[width=0.40\textwidth]{./fig/exemploFig1}
\caption{Uma figura típica.}
\label{fig:exemploFig1}
\end{figure}
```

Ao se usar o compilador \LaTeX , as figuras podem estar nos formatos *eps* e *ps*. Ao se usar o \PDFLaTeX , as figuras podem estar nos formatos *png*, *jpg*, *pdf* e *mps*. A classe `graphicx` também pode ser usada para a inclusão de figuras, nos formatos listados, ao se usar o \PDFLaTeX . Os comandos necessários são os mesmos ao se incluir figuras ao se usar o compilador \LaTeX . O uso do comando `\includegraphics` faz com que \PDFLaTeX procure primeiro por figuras com extensão *pdf*, depois *jpg*, depois *mps* e por último *png*. Aqui também não é necessário especificar a extensão do arquivo.

Para a inclusão das figuras 6.1 à 6.3 os comandos usados, tanto no \LaTeX quanto no \PDFLaTeX , seriam os mesmos. É claro que em cada caso devem estar disponíveis as figuras nos formatos suportados por cada compilador. Por exemplo, para a inclusão da figura 6.3 foram usados:

```

\begin{figure}[H]
\centering
\includegraphics[width=0.40\textwidth]{./fig/exemploFig3}
\caption{Figura incluída no texto com a classe graphicx.}
\label{fig:exemploFig3}
\end{figure}

```

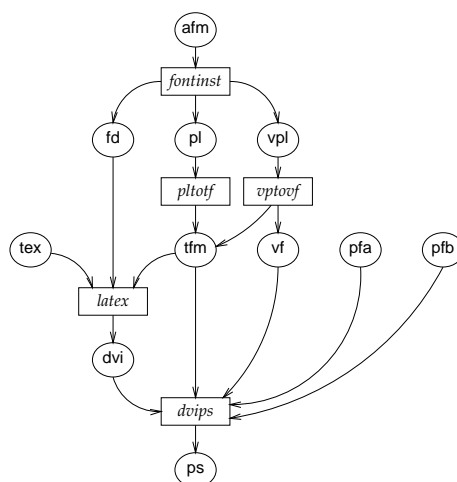


Figura 3.1: Uma figura típica.

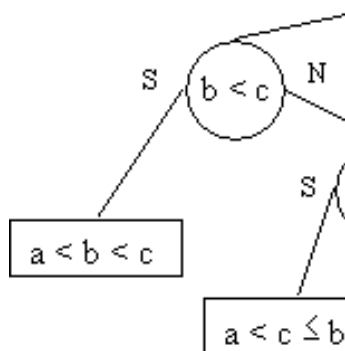


Figura 3.2: Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.

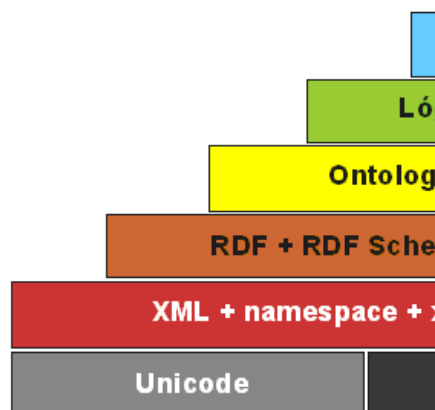
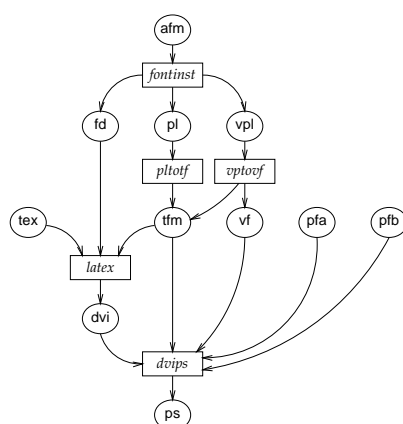


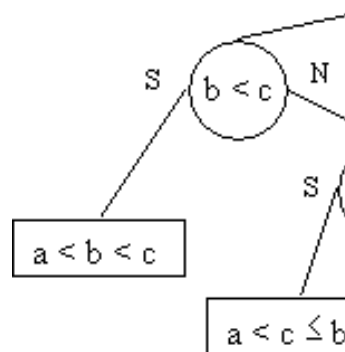
Figura 3.3: Figura incluída no texto com a classe `graphicx`.

3.1.1 Subfiguras

A classe `subfigure` pode ser usada para a inclusão de figuras dentro de figuras (consulte a documentação da classe para maiores detalhes). Por exemplo, a Figura 6.4 contém duas subfiguras. Estas podem ser referenciadas por rótulos independentes, ou seja, podem ser referenciadas como Figuras 6.4(a) e 6.4(b) ou Subfiguras (a) e (b).



(a) Primeira subfigura.



(b) Segunda subfigura (um pedaço).

Figura 3.4: (a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.

A figura 6.4 foi incluída com os comandos listados a seguir. Observe que há rótulos independentes para cada uma das subfiguras e um rótulo geral para a figura, os quais podem ser todos referenciados.

```
\begin{figure}[h]
\centering
\subfigure[Primeira subfigura.]
{
\includegraphics[width=0.35\textwidth]{./fig/exemploFig1}
\label{subfig:ex1}
} \quad
\subfigure[Segunda subfigura (um pedaço).]
{
\includegraphics[width=0.30\textwidth]{./fig/exemploFig2}
\label{subfig:ex2}
}
\caption{{\subref{subfig:ex1}} e {\subref{subfig:ex2}} representam
dois exemplos do uso de subfiguras dentro de uma única
figura.}
\label{fig:subfiguras}
\end{figure}
```

Caso uma subfiguras não tenha rótulo, para evitar que o apenas o número da mesma apareça na Lista de Figuras, use o comando `\subfigure[] []`. Caso uma subfigura tenha rótulo e deseja-se evitar que a mesma apareça na Lista de Figuras, use o comando `\subfigure[] [Rótulo]`.

3.2 Tabelas

Em tabelas, deve-se evitar usar cor de fundo diferente do branco e o uso de linhas grossas ou duplas. Ao relatar dados empíricos, não se deve usar mais dígitos decimais do aqueles que possam ser garantidos pela sua precisão e reprodutibilidade. Rótulos de tabelas devem ser colocados antes das mesmas (veja a Tabela 6.1).

3.3 Algoritmos

Algoritmos devem ser representados no formato do Algoritmo 6.1, que foi descrito com o uso da classe `algorithm2e`. A rigor não é obrigatório o uso dessa classe,

contudo o uso da mesma permite que seja gerada automaticamente uma lista de algoritmos logo após o sumário.

Algoritmo 3.1: $MSR(A, i, j)$

Entrada: vetor $A[i..j]$, inteiros não negativos i e j .

Saída: vetor $A[i..j]$ ordenado.

```
1  $n \leftarrow j - i$ .
2 se  $(n < 4)$  então
3   | Ordene com  $\leq 3$  comparações.
4 senão
5   | Divida  $A$  em  $\lceil \sqrt{n} \rceil$  subvetores de comprimento máximo  $\lfloor \sqrt{n} \rfloor$ .
6   | Aplique  $MSR$  a cada um dos subvetores.
7   | Intercale os subvetores.
8 fim
```

3.4 Códigos de Programa

Códigos de programa podem ser importados, mantendo-se a formatação original, conforme se pode ver no exemplo do Código 6.1. Este exemplo usa o ambiente `codigo`, definido na classe `inf-ufg`, que permite que uma lista de programas seja gerada automaticamente logo após o sumário.

Código 3.1 `insertionsort()`

```

1 void insertionSort( int* v, int n )
2 {
3     int i    = 0;
4     int j    = 1;
5     int aux = 0;
6
7     while (j < n)
8     {
9         aux = v[j];
10        i   = j - 1;
11        while ((i >= 0) && (v[i] > aux))
12        {
13            v[i + 1] = v[i];
14            i = i - 1;
15        }
16        v[i + 1] = aux;
17        j = j + 1;
18    }
19 }

```

3.5 Teoremas, Corolários e Demonstrações

O uso do ambiente `theorem` permite a escrita de teoremas, como no exemplo a seguir:

```
\begin{theorem}[Pitágoras]
```

Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.

```
\end{theorem}
```

O resultado é o mostrado a seguir:

Teorema 3.1 (Pitágoras) *Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.*

Da mesma forma pode-se usar o ambiente `proof` para demonstrações de teoremas:

```
\begin{proof}
```

Para demonstrar o Teorema de Pitágoras \dots

```
\end{proof}
```

Neste caso, o resultado é:

Prova. Para demonstrar o Teorema de Pitágoras ...

□

Além desses dois ambientes, estão definidos os ambientes `definition` (Definição), `corollary` (Corolário), `lemma` (Lema), `proposition` (Proposição), `comment` (Observação).

3.6 Citações Longas

Segundo as normas da ABNT, uma citação longa (mais de 3 linhas) deve seguir uma formação especial. Para tanto foi criado o ambiente `citacao`, o qual é baseado no ambiente de mesmo nome definido pelo grupo ABNTex [?]:

Uma citação longa (mais de 3 linhas) deve vir em parágrafo separado, com recuo de 4cm da margem esquerda, em fonte menor, sem as aspas [?, 4.4] e com espaçamento simples [?, 5.3]. Uma regra de como fazer citações em geral não é simples. É prudente ler [?] se você optar for fazer uso freqüente de citações. Para satisfazer às exigências tipográficas que a norma pede para citações longas, use o ambiente `citacao`.

Este exemplo de citação longa foi produzido com o uso do ambiente `citacao`, como descrito logo a seguir:

```
\begin{citacao}
Uma citação longa (mais de 3 linhas) deve vir em parágrafo
separado, com recuo de 4cm da margem esquerda, em fonte menor,
sem as aspas \cite[4.4]{NBR10520:2001} e com espaçamento
simples \cite[5.3]{NBR14724:2001}. Uma regra de como fazer
citações em geral não é simples. É prudente ler
\cite{NBR10520:2001} se você optar for fazer uso freqüente
de citações. Para satisfazer às exigências tipográficas que a
norma pede para citações longas, use o ambiente citacao.
\end{citacao}
```

3.7 Referências Bibliográficas

Esta seção mostra exemplos de uso de referências bibliográficas com `BIBTEX` e do comando `\cite`. Muitas das entradas listadas na página 44 foram obtidas de: <http://liinwww.ira.uka.de/bibliography/index.html>. Outro grande repositório de referências já em formato `BIBTEX` está disponível em: <http://www.math.utah.edu/bebe/bibliographies.html>.

As referências bibliográficas devem ser não ambíguas e uniformes. Recomenda-se usar números entre colchetes, como por exemplo [?], [?] e [?]. O comando `\nocite` não produz texto, mas permite que a entrada seja incluída nas referências. Por exemplo, o comando `\nocite{Ber1970}` gera na lista de referências bibliográficas a entrada referente à chave `Ber1970`, mas não inclui nenhuma referência no texto. O comando `\nocite{*}` faz com que todas as entradas do arquivo de dados do `BIBTEX` sejam incluídas nas referências.

Existem vários livros sobre `LATEX`, como [?, ?, ?], embora os mais famosos sejam sem dúvida [?] e [?]. Para converter documentos `LATEX` para HTML veja [?, pg.1–10].

Tabela 3.1: *Conteúdo do diretório [?]*

Tag	Comprimento	Início		Tag	Comprimento	Início
001	0020	00000		100	0032	00235
003	0004	00020		245	0087	00267
005	0017	00024		246	0036	00354
008	0041	00041		250	0012	00390
010	0024	00082		260	0037	00402
020	0025	00106		300	0029	00439
020	0044	00131		500	0042	00468
040	0018	00175		520	0220	00510
050	0024	00193		650	0033	00730
082	0018	00217		650	0012	00763

Introdução

Este documento mostra como usar o \LaTeX com a classe `inf-ufg` para formatar teses, dissertações, monografias e relatórios de conclusão de curso, segundo o padrão adotado pelo Instituto de Informática da UFG. Este documento e a classe `inf-ufg` foram, em grande parte, copiados e adaptados da classe `thesisPUC` e do texto de Thomas Lewiner [?] que descreve a sua utilização.

\LaTeX é um sistema de editoração eletrônica muito usado para produzir documentos científicos de alta qualidade tipográfica. O sistema também é útil para produzir todos os tipos de outros documentos, desde simples cartas até livros completos.

Se você precisar de algum material de apoio referente ao \LaTeX , dê uma olhada em um dos sites do Comprehensive TEX Archive Network (CTAN). O site está em www.ctan.org. Todos os pacotes podem ser obtidos via FTP <ftp://www.ctan.org> e existem vários servidores em todo o mundo. Eles podem ser encontrados, por exemplo, em <ftp://ctan.tug.org> (EUA), <ftp://ftp.dante.de> (Alemanha), <ftp://ftp.tex.ac.uk> (Reino Unido).

Você pode encontrar uma grande quantidade de informações e dicas na página dos usuários brasileiros de \LaTeX (\TeX -BR). O endereço é <http://biquinho.furg.br/tex-br/>. Tanto no CTAN quanto no \TeX -BR estão disponíveis bons documentos em português sobre o \LaTeX . Em particular no CTAN, está disponível uma introdução bastante completa em português: [CTAN:/tex-archive/info/lshort/portuguese-BR/](http://www.ctan.org/tex-archive/info/lshort/portuguese-BR/). No \TeX -BR também existe um documento com exemplos de uso de \LaTeX e de vários pacotes: <http://biquinho.furg.br/tex-br/doc/LaTeX-demo/>. O objetivo é ser, através de exemplos, um guia para o usuário de \LaTeX iniciante e intermediário, podendo, ainda, servir como um guia de referência rápida para usuários avançados.

Se você quer usar o \LaTeX em seu computador, verifique em quais sistemas ele está disponível em [CTAN:/tex-archive/systems](http://www.ctan.org/tex-archive/systems). Em particular para MS Windows, o sistema gratuito **MikTeX**, disponível no CTAN e no site www.miktex.org é completo e atualizado de todas as opções que você poderia precisar para editar o seu texto.

O estilo `inf-ufg` se integra completamente ao \LaTeX 2 ϵ . Uma tese, dissertação ou monografia escrita no estilo padrão do \LaTeX para teses (estilo `report`) pode ser formatada em 15 minutos para se adaptar às normas da UFG.

O estilo inf-ufg foi desenhado para minimizar a quantidade de texto e de comandos necessários para escrever a sua dissertação. Só é preciso inserir algumas macros no início do seu arquivo \LaTeX , precisando os dados bibliográficos da sua dissertação (por exemplo o seu nome, o título da dissertação...). Em seguida, cada página dos elementos pré-textuais será formatada usando macros ou ambientes específicos. O corpo do texto é editado normalmente. Finalmente, as referências bibliográficas podem ser entradas manualmente (via o comando `\bibitem` do \LaTeX padrão) ou usando o sistema BiBTeX (muito mais recomendável). Neste caso, os arquivos `inf-ufg.bst` e `abnt-alf.bst` permitem a formatação das referências bibliográficas segundo as normas da UFG.

Descrição da classe inf-ufg

5.1 Opções da classe

Para usar esta classe num documento $\text{\LaTeX} 2_{\epsilon}$, coloque os arquivos `inf-ufg.cls`, `inf-ufg.bst`, `abnt-num.bst`, `atbeginend.sty` e `tocloft.sty` numa pasta onde o compilador \LaTeX pode achá-lo (normalmente na mesma pasta que seu arquivo `.tex`), e defina-o como o estilo do seu documento. Por exemplo, uma dissertação de mestrado que usa o modelo abnt de citações bibliográficas:

```
\documentclass[dissertacao,abnt]{inf-ufg}
...
\begin{document}
```

As opções da classe são `[tese]` (para tese de doutorado), `[dissertacao]` (para dissertação de mestrado), `[monografia]` (para monografia de curso de especialização e `[relatorio]` (para relatório final de curso de graduação). Se nenhuma opção for declarada, o documento é considerado como uma dissertação de mestrado. Se a opção `[abnt]` for utilizada, as citações bibliográficas serão geradas conforme definido pelo grupo de trabalho `abnt-tex`. Contudo, o mais recomendável é não utilizar essa opção. Com a opção `[nocolorlinks]` todos os *links* de navegação no texto ficam na cor preta. O ideal é usar esta opção para gerar o arquivo para impressão, pois a qualidade da impressão dos *links* fica superior.

5.2 Parâmetros da classe

Os elementos pré-textuais são definidos página por página e dependem da correta definição dos parâmetros listados a seguir (aqueles que contém um texto/valor padrão não precisam ser definidos, caso atenda a situação do autor do texto que está usando a classe `inf-ufg.cls`):

- `\autor` : Nome completo do autor da tese, começando pelo apelido (ex.: José da Silva);

- `\autorR` : Nome completo do autor da tese, começando pelo nome (ex.: da Silva, José);
- `\titulo` : Título da tese, dissertação, monografia ou relatório de conclusão de curso;
- `\subtitulo` : Se tiver um subtítulo, use este macro para defini-lo;
- `\cidade` : A cidade de edição. A cidade padrão é Goiânia.
- `\dia` : Dia do mês da data de defesa (1–31);
- `\mes` : Mês da data de defesa (1–12);
- `\ano` : Ano da data de defesa;
- `\universidade` : Nome completo da universidade. O nome padrão é Universidade Federal de Goiás;
- `\uni` : Sigla da universidade. A sigla padrão é UFG;
- `\unidade` : Nome da unidade acadêmica. O padrão é Instituto de Informática;
- `\departamento` : Nome do departamento, com maiúscula na primeira letra (para o caso de unidades com mais de um departamento);
- `\programa` : Nome do programa de pós-graduação, com maiúscula na primeira letra. O padrão é Computação;
- `\concentracao` : Nome da área de concentração;
- `\orientador` : Nome completo do orientador, começando pelo apelido;
- `\orientadorR` : Nome completo do orientador, começando pelo nome;
- `\orientadora` : Nome completo da orientadora, começando pelo apelido; use este comando e o próximo se for orientadora e não orientador.
- `\orientadoraR` : Nome completo do orientadora, começando pelo nome;
- `\coorientador` : Nome completo do co-orientador, começando pelo apelido;
- `\coorientadorR` : Nome completo do co-orientador, começando pelo nome;
- `\coorientadora` : Nome completo da coorientadora, começando pelo apelido; use este comando e o próximo se for coorientadora e não coorientador.
- `\coorientadoraR` : Nome completo do coorientadora, começando pelo nome;
- `\universidadeco` : Nome da universidade do coorientador;
- `\unico` : Sigla da universidade do coorientador;
- `\unidadeco` : Nome da unidade acadêmica do coorientador.¹

5.3 Elementos Pré-Textuais

Os elementos pré-textuais são definidos página por página, conforme descritos a seguir:

¹ Se não tiver um co-orientador, não defina esses últimos sete parâmetros.

capa

`\capa` : Gera o modelo da capa externa do trabalho. Esta página servirá apenas como modelo para a encadernação da versão final do texto. Nenhum dado é necessário.

publicação

`\publica` : Gera a autorização para publicação do trabalho em formato eletrônico e disponibilização do mesmo na biblioteca virtual da UFG.

rosto

`\rosto` : Gera a folha de rosto, a qual é a primeira folha interna do trabalho. Nenhum dado é necessário.

aprovação

`\aprovacao` : ambiente para a reprodução do termo de aprovação da Banca Examinadora da tese ou dissertação.

banca

`\banca` : Entrada para o nome dos examinadores, exceto o(s) orientador(es).

`\profa` : Entrada para o nome das examinadoras, exceto o(s) orientador(es).

direitos

`\direitos` : Macro com 2 argumentos para gerar os direitos autorais, o perfil do aluno e a ficha catalográfica da Biblioteca Central da UFG.

- O primeiro argumento é o Perfil do aluno; e
- O segundo argumento é a lista das palavras-chaves para a Ficha Catalográfica.

dedicatória

`\dedicatoria` : ambiente para escrever a dedicatória. É possível trocar o espaçamento dentro desse ambiente do mesmo jeito que no \LaTeX padrão.

agradecimentos

`\agradecimentos` : ambiente para escrever os agradecimentos. É possível trocar o espaçamento dentro desse ambiente do mesmo jeito que no \LaTeX padrão.

resumo

`\chaves` : A lista das palavras chaves, separadas por ‘;’. Deve ser definido antes do ambiente `\resumo`, o qual é usado para escrever o resumo em português.

abstract

`\keys` : A lista das palavras chaves em inglês, separadas por ‘;’. Deve ser definido antes do ambiente `\abstract`, o qual contém 1 argumento e é usado escrever o resumo em inglês. O argumento deve ser o título do trabalho em inglês.

tabelas

`\tabelas` : Macro com 1 argumento opcional para gerar as tabelas. O argumento pode ser:

- nada [] : gera apenas o sumário;
- fig : gera o sumário e uma lista de figuras;
- tab : gera o sumário e uma lista de tabelas;
- alg : gera o sumário e uma lista de algoritmos;
- cod : gera o sumário e uma lista de programas.

Pode-se usar qualquer combinação dessas opções. Por exemplo:

- figtab : gera o sumário e listas de figuras e tabelas,
- figtabcod : gera o sumário e listas de figuras, tabelas e códigos de programas;
- figtabalg : gera o sumário e listas de figuras, tabelas e algoritmos;
- figtabalgcod : gera o sumário e listas de figuras, tabelas, algoritmos e códigos de programas

epígrafe

`\epigrafe` : Macro com 3 argumentos que permite editar um epígrafe. O primeiro argumento é o texto da citação. O segundo argumento é o nome do autor da citação. O terceiro argumento é o título da referência à qual a citação pertence.

Elementos do texto

6.1 Figuras

Rótulos de figuras e tabelas devem ser centralizados se tiverem até uma linha (Figura 6.1), caso contrário devem estar justificados e identados em ambas as margens, como mostrado na Figura 6.2. Essa formatação já é realizada automaticamente pela classe `inf-ufg`.

Os compiladores \LaTeX provêem um mecanismo bastante simples para inclusão de figuras, o que pode ser feito com o auxílio de várias classes auxiliares (as mais comuns são `graphic` e `graphicx`). A classe `inf-ufg` usa o comando `\includegraphics`, da classe `graphicx`, para a inclusão de figuras e não é necessário você colocar a extensão do arquivo neste comando. Por exemplo, para a figura 6.1 os comandos usados foram:

```
\begin{figure}[htb]
\centering
\includegraphics[width=0.40\textwidth]{./fig/exemploFig1}
\caption{Uma figura típica.}
\label{fig:exemploFig1}
\end{figure}
```

Ao se usar o compilador \LaTeX , as figuras podem estar nos formatos *eps* e *ps*. Ao se usar o \PDFLaTeX , as figuras podem estar nos formatos *png*, *jpg*, *pdf* e *mps*. A classe `graphicx` também pode ser usada para a inclusão de figuras, nos formatos listados, ao se usar o \PDFLaTeX . Os comandos necessários são os mesmos ao se incluir figuras ao se usar o compilador \LaTeX . O uso do comando `\includegraphics` faz com que \PDFLaTeX procure primeiro por figuras com extensão *pdf*, depois *jpg*, depois *mps* e por último *png*. Aqui também não é necessário especificar a extensão do arquivo.

Para a inclusão das figuras 6.1 à 6.3 os comandos usados, tanto no \LaTeX quanto no \PDFLaTeX , seriam os mesmos. É claro que em cada caso devem estar disponíveis as figuras nos formatos suportados por cada compilador. Por exemplo, para a inclusão da figura 6.3 foram usados:

```

\begin{figure}[H]
\centering
\includegraphics[width=0.40\textwidth]{./fig/exemploFig3}
\caption{Figura incluída no texto com a classe graphicx.}
\label{fig:exemploFig3}
\end{figure}

```

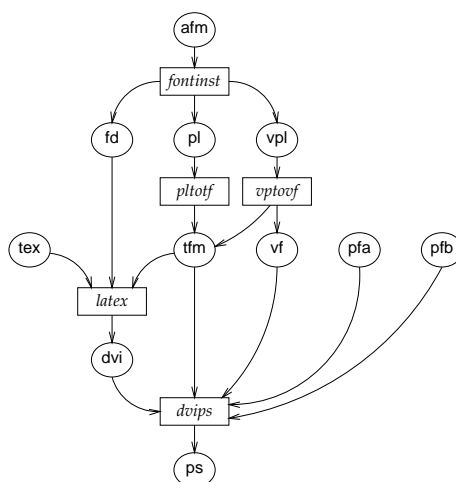


Figura 6.1: Uma figura típica.

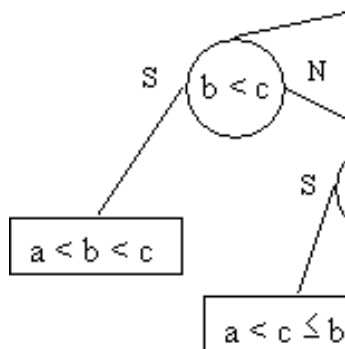


Figura 6.2: Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.

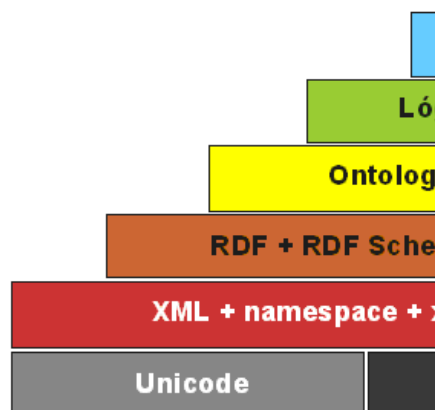
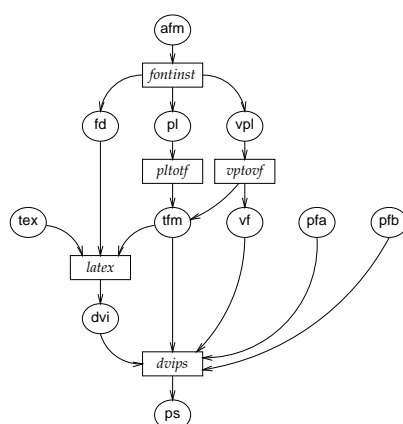


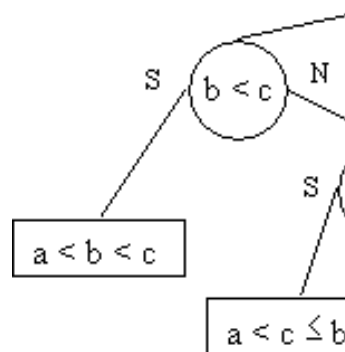
Figura 6.3: Figura incluída no texto com a classe `graphicx`.

6.1.1 Subfiguras

A classe `subfigure` pode ser usada para a inclusão de figuras dentro de figuras (consulte a documentação da classe para maiores detalhes). Por exemplo, a Figura 6.4 contém duas subfiguras. Estas podem ser referenciadas por rótulos independentes, ou seja, podem ser referenciadas como Figuras 6.4(a) e 6.4(b) ou Subfiguras (a) e (b).



(a) Primeira subfigura.



(b) Segunda subfigura (um pedaço).

Figura 6.4: (a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.

A figura 6.4 foi incluída com os comandos listados a seguir. Observe que há rótulos independentes para cada uma das subfiguras e um rótulo geral para a figura, os quais podem ser todos referenciados.

```
\begin{figure}[h]
\centering
\subfigure[Primeira subfigura.]
{
\includegraphics[width=0.35\textwidth]{./fig/exemploFig1}
\label{subfig:ex1}
} \quad
\subfigure[Segunda subfigura (um pedaço).]
{
\includegraphics[width=0.30\textwidth]{./fig/exemploFig2}
\label{subfig:ex2}
}
\caption{{\subref{subfig:ex1}} e {\subref{subfig:ex2}} representam
dois exemplos do uso de subfiguras dentro de uma única
figura.}
\label{fig:subfiguras}
\end{figure}
```

Caso uma subfiguras não tenha rótulo, para evitar que o apenas o número da mesma apareça na Lista de Figuras, use o comando `\subfigure[] []`. Caso uma subfigura tenha rótulo e deseja-se evitar que a mesma apareça na Lista de Figuras, use o comando `\subfigure[] [Rótulo]`.

6.2 Tabelas

Em tabelas, deve-se evitar usar cor de fundo diferente do branco e o uso de linhas grossas ou duplas. Ao relatar dados empíricos, não se deve usar mais dígitos decimais do aqueles que possam ser garantidos pela sua precisão e reprodutibilidade. Rótulos de tabelas devem ser colocados antes das mesmas (veja a Tabela 6.1).

6.3 Algoritmos

Algoritmos devem ser representados no formato do Algoritmo 6.1, que foi descrito com o uso da classe `algorithm2e`. A rigor não é obrigatório o uso dessa classe,

contudo o uso da mesma permite que seja gerada automaticamente uma lista de algoritmos logo após o sumário.

Algoritmo 6.1: $MSR(A, i, j)$

Entrada: vetor $A[i..j]$, inteiros não negativos i e j .

Saída: vetor $A[i..j]$ ordenado.

```
1  $n \leftarrow j - i$ .
2 se ( $n < 4$ ) então
3   | Ordene com  $\leq 3$  comparações.
4 senão
5   | Divida  $A$  em  $\lceil \sqrt{n} \rceil$  subvetores de comprimento máximo  $\lfloor \sqrt{n} \rfloor$ .
6   | Aplique  $MSR$  a cada um dos subvetores.
7   | Intercale os subvetores.
8 fim
```

6.4 Códigos de Programa

Códigos de programa podem ser importados, mantendo-se a formatação original, conforme se pode ver no exemplo do Código 6.1. Este exemplo usa o ambiente `codigo`, definido na classe `inf-ufg`, que permite que uma lista de programas seja gerada automaticamente logo após o sumário.

Código 6.1 insertionsort()

```

1 void insertionSort( int* v, int n )
2 {
3     int i    = 0;
4     int j    = 1;
5     int aux = 0;
6
7     while (j < n)
8     {
9         aux = v[j];
10        i   = j - 1;
11        while ((i >= 0) && (v[i] > aux))
12        {
13            v[i + 1] = v[i];
14            i = i - 1;
15        }
16        v[i + 1] = aux;
17        j = j + 1;
18    }
19 }

```

6.5 Teoremas, Corolários e Demonstrações

O uso do ambiente `theorem` permite a escrita de teoremas, como no exemplo a seguir:

```
\begin{theorem}[Pitágoras]
```

Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.

```
\end{theorem}
```

O resultado é o mostrado a seguir:

Teorema 6.1 (Pitágoras) *Em todo triângulo retângulo o quadrado do comprimento da hipotenusa é igual a soma dos quadrados dos comprimentos dos catetos.*

Da mesma forma pode-se usar o ambiente `proof` para demonstrações de teoremas:

```
\begin{proof}
```

Para demonstrar o Teorema de Pitágoras \dots

```
\end{proof}
```

Neste caso, o resultado é:

Prova. Para demonstrar o Teorema de Pitágoras ...

□

Além desses dois ambientes, estão definidos os ambientes `definition` (Definição), `corollary` (Corolário), `lemma` (Lema), `proposition` (Proposição), `comment` (Observação).

6.6 Citações Longas

Segundo as normas da ABNT, uma citação longa (mais de 3 linhas) deve seguir uma formação especial. Para tanto foi criado o ambiente `citacao`, o qual é baseado no ambiente de mesmo nome definido pelo grupo ABNTEX [?]:

Uma citação longa (mais de 3 linhas) deve vir em parágrafo separado, com recuo de 4cm da margem esquerda, em fonte menor, sem aspas [?, 4.4] e com espaçamento simples [?, 5.3]. Uma regra de como fazer citações em geral não é simples. É prudente ler [?] se você optar por fazer uso freqüente de citações. Para satisfazer às exigências tipográficas que a norma pede para citações longas, use o ambiente `citacao`.

Este exemplo de citação longa foi produzido com o uso do ambiente `citacao`, como descrito logo a seguir:

```
\begin{citacao}
Uma citação longa (mais de 3 linhas) deve vir em parágrafo
separado, com recuo de 4cm da margem esquerda, em fonte menor,
sem aspas \cite[4.4]{NBR10520:2001} e com espaçamento
simples \cite[5.3]{NBR14724:2001}. Uma regra de como fazer
citações em geral não é simples. É prudente ler
\cite{NBR10520:2001} se você optar por fazer uso freqüente
de citações. Para satisfazer às exigências tipográficas que a
norma pede para citações longas, use o ambiente citacao.
\end{citacao}
```

6.7 Referências Bibliográficas

Esta seção mostra exemplos de uso de referências bibliográficas com `BIBTEX` e do comando `\cite`. Muitas das entradas listadas na página 44 foram obtidas de: <http://liinwww.ira.uka.de/bibliography/index.html>. Outro grande repositório de referências já em formato `BIBTEX` está disponível em: <http://www.math.utah.edu/bebe/bibliographies.html>.

As referências bibliográficas devem ser não ambíguas e uniformes. Recomenda-se usar números entre colchetes, como por exemplo [?], [?] e [?]. O comando `\nocite` não produz texto, mas permite que a entrada seja incluída nas referências. Por exemplo, o comando `\nocite{Ber1970}` gera na lista de referências bibliográficas a entrada referente à chave `Ber1970`, mas não inclui nenhuma referência no texto. O comando `\nocite{*}` faz com que todas as entradas do arquivo de dados do `BIBTEX` sejam incluídas nas referências.

Existem vários livros sobre `LATEX`, como [?, ?, ?], embora os mais famosos sejam sem dúvida [?] e [?]. Para converter documentos `LATEX` para HTML veja [?, pg.1–10].

Tabela 6.1: *Conteúdo do diretório [?]*

Tag	Comprimento	Início		Tag	Comprimento	Início
001	0020	00000		100	0032	00235
003	0004	00020		245	0087	00267
005	0017	00024		246	0036	00354
008	0041	00041		250	0012	00390
010	0024	00082		260	0037	00402
020	0025	00106		300	0029	00439
020	0044	00131		500	0042	00468
040	0018	00175		520	0220	00510
050	0024	00193		650	0033	00730
082	0018	00217		650	0012	00763

Referências Bibliográficas

- [1] DELAMARO, M. E; MALDONADO, J. C. J. M. **Teste de mutação**. In: *Introducao ao Teste de Software*, p. 77–118, Rio de Janeiro, 2007. Elsevier.
- [2] ELMASRI, RAMEZ; NAVATHE, S. **Sistemas de banco de dados**. São Paulo, sixth edition, 2005.
- [3] KORTH, HENRY F.; SILBERSCHATZ, A. S. S. **Sistema de banco de dados**. Makron Books, São Paulo, 1999.
- [4] MODESTO, L. R. **Teste funcional baseado em diagramas da uml**. Dissertação de mestrado em ciência da computação, Centro Universitário Eurípides de Marília (UNIVEM), Marília, São Paulo, Brasil, 2006.
- [5] MYERS, G. J. **The art of software testing**. John Wiley & Sons, Inc, Hoboken, New Jersey, 2004.
- [6] SOMMERVILLE, I. **Engenharia de software**. Pearson, São Paulo, 2008.
- [7] SOUZA, J. P. D. **Teste funcional em aplicação de banco de dados relacional baseado nos diagramas da uml**, 2008.
- [8] SPOTO, E. S. **Teste estrutural de programas de aplicação de banco de dados relacional**. Tese de doutorado em engenharia elétrica, Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas - FEEC/UNICAMP, Campinas, 2000.

Apêndicess são iniciados com o comando \apendices. Apêndicess são inicia-
dos com o comando \apendices. Apêndicess são iniciados com o comando \apendices.
Apêndicess são iniciados com o comando \apendices. Apêndicess são iniciados com o
comando \apendices. Apêndicess são iniciados com o comando \apendices. Apên-
dicess são iniciados com o comando \apendices. Apêndicess são iniciados com o co-
mando \apendices. Apêndicess são iniciados com o comando \apendices. Apêndi-
cess são iniciados com o comando \apendices. Apêndicess são iniciados com o co-
mando \apendices. Apêndicess são iniciados com o comando \apendices. Apêndicess
são iniciados com o comando \apendices. Apêndicess são iniciados com o comando
\apendices.

[illegible][illegible][illegible]

[illegible]

Apêndices são iniciados com o comando \apendices. Apêndices são iniciados
com o comando \apendices. Apêndices são iniciados com o comando \apendices.
Apêndices são iniciados com o comando \apendices. Apêndices são iniciados com o
comando \apendices. Apêndices são iniciados com o comando \apendices. Apêndi-
ces são iniciados com o comando \apendices. Apêndices são iniciados com o comando
\apendices. Apêndices são iniciados com o comando \apendices. Apêndices são inici-
ados com o comando \apendices. Apêndices são iniciados com o comando \apendices.
Apêndices são iniciados com o comando \apendices. Apêndices são iniciados com o co-
mando \apendices. Apêndices são iniciados com o comando \apendices.

[illegible]