

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 Relacional**. Goiânia, 2017. 47p. 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. 47p. 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
1 Introdução	13
1.1 Objetivos	13
1.2 Motivação	13
2 Teste de Software	14
2.1 Conceitos de Teste de Software	14
2.2 Técnicas e Critérios de Teste de Software	14
2.3 Teste Funcional	14
2.4 Teste de Mutação	14
3 Elementos do texto	15
3.1 Figuras	15
3.1.1 Subfiguras	17
3.2 Tabelas	18
3.3 Algoritmos	18
3.4 Códigos de Programa	19
3.5 Teoremas, Corolários e Demonstrações	20
3.6 Citações Longas	21
3.7 Referências Bibliográficas	21
4 Introdução	24
5 Descrição da classe inf-ufg	26
5.1 Opções da classe	26
5.2 Parâmetros da classe	26
5.3 Elementos Pré-Textuais	27
6 Elementos do texto	30
6.1 Figuras	30
6.1.1 Subfiguras	32
6.2 Tabelas	33
6.3 Algoritmos	33
6.4 Códigos de Programa	34
6.5 Teoremas, Corolários e Demonstrações	35
6.6 Citações Longas	36

6.7	Referências Bibliográficas	36
	Referências Bibliográficas	<b>39</b>
A	Exemplo de um Apêndice	<b>40</b>
B	Exemplo de Outro Apêndice	<b>44</b>

---

## Lista de Figuras

---

3.1	Uma figura típica.	16
3.2	Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.	16
3.3	Figura incluída no texto com a classe <code>graphicx</code> .	17
3.4	(a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.	17
	(b) Segunda subfigura (um pedaço).	17
6.1	Uma figura típica.	31
6.2	Esta figura é um exemplo de um rótulo de figura que ocupa mais de uma linha, devendo ser indentado e justificado.	31
6.3	Figura incluída no texto com a classe <code>graphicx</code> .	32
6.4	(a) e (b) representam dois exemplos do uso de subfiguras dentro de uma única figura.	32
	(b) Segunda subfigura (um pedaço).	32

---

# Lista de Tabelas

---

3.1	<a href="#">Conteúdo do diretório [?]</a>	23
6.1	<a href="#">Conteúdo do diretório [?]</a>	38

## **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.

### **1.1 Objetivos**

Objetivos

### **1.2 Motivação**

Motivacao

## Teste de Software

---

### 2.1 Conceitos de Teste de Software

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 Técnicas e Critérios de Teste de Software

### 2.3 Teste Funcional

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

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

### 2.4 Teste de Mutação

X-Man

### 2.5 Banco de Dados



---

## Banco de Dados

---

### 3.1 Restrições de Banco de Dados

#### 3.1.1 Restrições de Domínio

#### 3.1.2 Restrições de Integridade Semântica

#### 3.1.3 Restrições de Integridade Referencial e de Chave Primária

### 3.2 Aplicações Banco de Dados

### 3.3 Linguagem de Banco de Dados

### 3.4 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`.

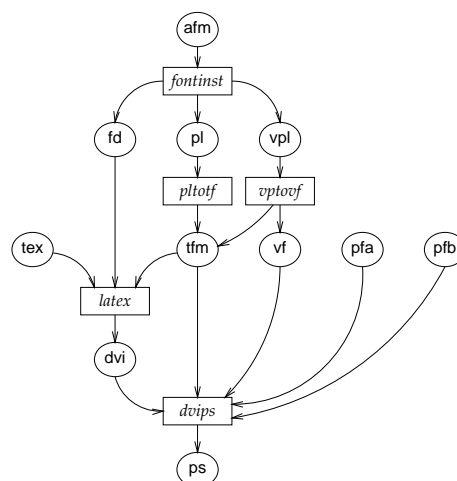
Os compiladores  $\text{\LaTeX}$  provêem um mecanismo bastante simples para inclusão de figuras, o que pode ser feito com o auxílio de várias classes auxiliares (as mais comuns são `graphic` e `graphicx`). A classe `inf-ufg` usa o comando `\includegraphics`, da classe `graphicx`, para a inclusão de 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}
```

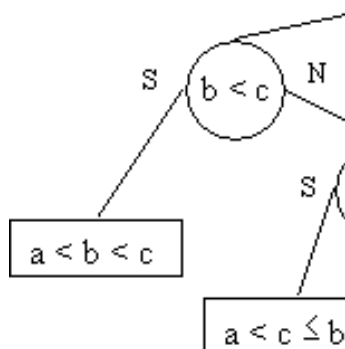
Ao se usar o compilador  $\text{\LaTeX}$ , as figuras podem estar nos formatos *eps* e *ps*. Ao se usar o  $\text{PDF}\text{\LaTeX}$ , as figuras podem estar nos formatos *png*, *jpg*, *pdf* e *mps*. A classe `graphicx` também pode ser usada para a inclusão de figuras, nos formatos listados, ao se usar o  $\text{PDF}\text{\LaTeX}$ . Os comandos necessários são os mesmos ao se incluir figuras ao se usar o compilador  $\text{\LaTeX}$ . O uso do comando `\includegraphics` faz com que  $\text{PDF}\text{\LaTeX}$  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  $\text{\LaTeX}$  quanto no  $\text{PDF}\text{\LaTeX}$ , 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:

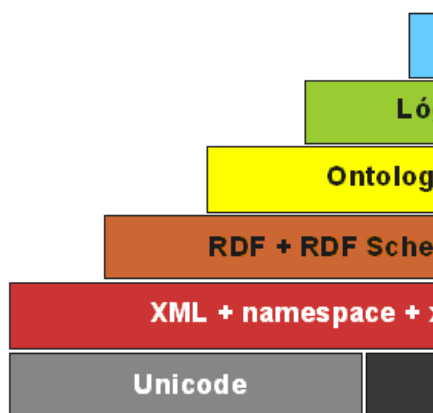
```
\begin{figure}[H]
\centering
\includegraphics[width=0.40\textwidth]{./fig/exemploFig3}
\caption{Figura incluída no texto com a classe graphicx.}
\label{fig:exemploFig3}
\end{figure}
```



**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.

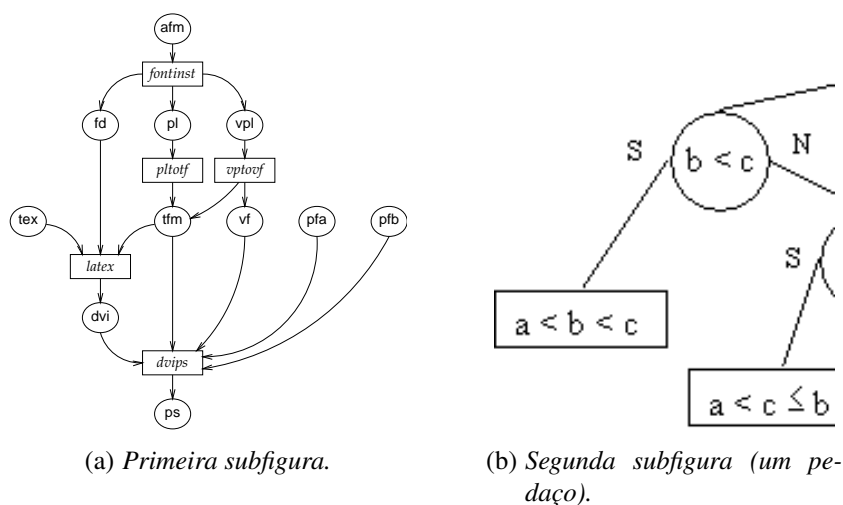


**Figura 3.3:** Figura incluída no texto com a classe `graphicx`.

### 3.4.1 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 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.



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

```
\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]`.

## 3.5 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).

**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

## 3.6 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
```

---

## 3.7 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 }
```

---

## 3.8 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).

## 3.9 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}
```

## 3.10 Referências Bibliográficas

Esta seção mostra exemplos de uso de referências bibliográficas com `BIBTEX` e do comando `\cite`. Muitas das entradas listadas na página 39 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].



## **Técnicas de Testes aplicadas em Banco de Dados Relacional**

---

### **4.1 Critérios de Teste Funcional em Aplicações de Banco de Dados**

#### **4.1.1 Critérios Para Testes Intra - Tabelas**

#### **4.1.2 Critérios Para Testes Inter-Tabelas**

### **4.2 Técnica de Teste Mutante em Banco de Dados**

## Estudo de Caso

---

## **Plano de Teste**

---

### **6.1 Plano de Testes Baseado no Teste Funcional em BDR**

#### **6.1.1 Testes Intra - Tabelas**

#### **6.1.2 Testes Inter - Tabelas**

## **Resultados obtidos**

---

---

## 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.

Apêndices são iniciados com o comando \apendices. Apêndices são iniciados com o comando \apendices. Apêndices são iniciados com o comando \apendices.

[illegible][illegible][illegible][illegible][illegible]

ces são iniciados com o comando \apendices. Apêndices são iniciados com o comando  
\apendices. Apêndices são iniciados com o comando \apendices. Apêndices são ini-  
ciados 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]

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.