

Estrutura Base para Trabalho de Conclusão de Curso de Ciência da Computação

Prof^o MSc. Jônatas Leite de Oliveira

Última Atualização: 30/09/2016

1) Introdução

A introdução é a parte inicial do texto, na qual devem constar a delimitação do assunto tratado, objetivos da pesquisa e outros elementos necessários para situar o tema do trabalho. Está estruturada da seguinte maneira:

- **Contexto**
 - *Deve-se iniciar o trabalho anunciando o assunto, dizendo a área em que ele se insere e introduzindo o tema de pesquisa. Isto deve ser bem detalhado no início do **capítulo 1 Introdução**.*
- **Motivação**
 - *Com a Motivação é preciso transformar o tema em problema (descrito no início da Introdução contextualizando), mostrando com maior clareza o que se pretende resolver ou de que forma se pretende contribuir com o trabalho de pesquisa realizado. A Motivação deve estar clara nas **seções 1.1 Problemática e suas subseções**.*
- **Objetivos**
 - *Já nos Objetivos deve-se **declarar explicitamente os objetivos gerais e específicos** que foram abordados durante a pesquisa realizada. **Seção 1.2 Objetivos e subseções**.*
 - *Link para melhor compreensão:*
<http://monografias.brasilecola.uol.com.br/regras-abnt/objetivos-gerais-objetivos-especificos.htm>
- **Metodologia**
 - *Na **seção 1.3 Metodologia**, deve-se apresentar de forma resumida e clara os passos empregados durante a pesquisa permitindo que o leitor compreenda como que partindo do problema declarado na motivação chegou-se nos resultados.*
 - *Link para melhor compreensão:*
<http://monografias.brasilecola.uol.com.br/regras-abnt/a-metodologia-na-pesquisa-cientifica-esmiucando-conceitos.htm>
- **Organização**
 - *Na **seção 1.4 Estrutura do Trabalho**, apresenta-se a estrutura da monografia declarando o conteúdo de cada capítulo. Por exemplo:*
 - *No Capítulo 2, o modo geral de como devem estar organizados os capítulos referentes à Fundamentação Teórica. O Capítulo 3 é referente ao Projeto, etc...*

Os itens Contexto, Motivação, Objetivos, Metodologia e Resultados (descritos no capítulo de Conclusão) deverão estar presentes no resumo de maneira clara e sucinta, formando um ÚNICO parágrafo.

1.1) Problematização

1.1.1) Formulação do Problema

1.1.2) Solução Proposta

1.2) Objetivos

1.2.1) Objetivo Geral

1.2.2) Objetivos Específicos

1.3) Metodologia

1.4) Estrutura do Trabalho

2) Fundamentação Teórica

- *A fundamentação teórica consiste na revisão de textos, artigos, livros, periódicos, enfim, todo o material pertinente à revisão da literatura que será utilizada quando da redação do trabalho. É o momento de ler, selecionar, interpretar e discutir o material da pesquisa.*
- *A fundamentação teórica consiste em embasar por meio das ideias de outros autores aspectos teóricos de sua pesquisa. De acordo com Mello (2006, p. 86), “a fundamentação teórica apresentada deve servir de base para a análise e interpretação dos dados coletados na fase de elaboração do relatório final. Dessa forma, os dados apresentados devem ser interpretados à luz das teorias existentes”.*

3) Desenvolvimento

- *Neste capítulo deve ser documentada a Arquitetura e Modelagem do Sistema, bem como suas principais características. Neste capítulo devem ser lembrados os Objetivos descritos na **seção 1.2 Objetivos**, tendo como objetivos explanar e detalhar o desenvolvimento da solução para o problema proposto na **seção 1.1 Problematização**.*

3.1) Projeto

3.1.1) Análise de Requisitos

- **Lista de Requisitos Funcionais**
 - *Deve-se listar de forma enumerada cada um dos requisitos funcionais identificados e existentes no sistema.*

- **Especificação dos Requisitos Funcionais**
 - *Após listados os requisitos funcionais, cada um deles deve ser detalhado de maneira que seja possível identificar os seguintes aspectos de cada requisito:*
 - *Situação: se está implementado ou não, bem como o percentual, se implementado, ou Validado no caso de 100% implementado e conferido com os objetivos;*
 - *Importância: baixa, média, crítica, onde baixa é considerado uma funcionalidade dispensável para o objetivo principal do sistema, e crítico uma funcionalidade principal.*
 - *Razão: caso não implementado ou implementado parcialmente, qual a razão.*
- **Lista de Requisitos Não-Funcionais**
 - *Idem a Lista de Requisitos Funcionais*
- **Especificação dos Requisitos Não-Funcionais**
 - *Idem a Especificação dos Requisitos Funcionais.*

Link sobre Engenharia de Requisitos: http://ricardobarcelar.com.br/aulas/eng_sw/mod2-eng_requisitos.pdf

3.1.2) Modelagem UML

- **Lista de Casos de Uso**
 - *Segundo Ivan Jacobson , podemos dizer que um caso de uso é um "documento narrativo que descreve a sequência de eventos de um ator que usa um sistema para completar um processo".*
 - *Para melhor organização, um sistema deve possuir uma Lista de Casos de Uso, onde nessa lista encontramos: Número do Caso de Uso, Nome, Ator e Breve Descrição do caso de uso. Podendo ser representada por uma Tabela.*
- **Diagrama de Casos de Uso**
 - *Nesta seção pode-se inserir um Diagrama Geral de Casos de Uso, contendo todos os Atores e todos os Casos de Uso com seus devidos relacionamentos. Por vezes este diagrama pode ficar grande, assim, para o trabalho final, se preciso, faça a impressão em uma folha A3 e a insira devidamente dobrada no trabalho. Nesta seção deve-se também falar de maneira geral sobre os Atores do sistema e suas devidas competências.*
- **Especificação de Casos de Uso**
 - *Após listados todos os Casos de Uso e de explanado o Diagrama Geral de Casos de Uso, é hora de especificar um por um. Para isso deve-se criar um modelo de Documento de Especificação de Caso de Uso. Para cada caso de uso a ser descrito é importante que seu diagrama seja inserido, contendo suas extensões e inclusões, bem como todos seus Atores.*
 - *Para melhor entendimento sobre Caso de Uso acesse os Links abaixo:*
 - <http://www.dsc.ufcg.edu.br/~sampaio/cursos/2007.1/Graduacao/SI-II/Uml/diagramas/usecases/usecases.htm>
 - http://desenvolvimento.dataprev.gov.br/visualizar_artefato.php?idprocesso=1&idartefato=59

- **Diagrama de Classes**

- Neste ponto você deve inserir o Diagrama de Classes do sistema, contendo todas as classes e seus devidos atributos e relacionamentos. Veja mais no link abaixo:
 - <http://docente.ifrn.edu.br/qivanaldorocho/disciplinas/engenharia-de-software-licenciatura-em-informatica/diagrama-de-classes>
 - http://ricardobarcelar.com.br/aulas/enq_sw/mod3-uml.pdf
- Assim como o diagrama de Casos de Uso, para melhor visibilidade imprima o Diagrama de Classes em uma folha A3.
- Para a criação dos Diagramas utilize a ferramenta Astah Community.

3.2) Implementação

- Neste capítulo é hora de mostrar o show, mostrar o que você produziu nas madrugadas em que ficou programando!

3.2.1) Execução do Sistema

- **Testes e Validações**
 - Aqui deve ser mostrado o sistema funcionando, suas funcionalidades, passo a passo, tire print das telas e os explique detalhadamente aqui, fazendo referência aos Requisitos Funcionais descritos na **seção 3.1.1** e validando os mesmos. Se necessário destaque áreas das imagens para melhor entendimento do leitor.

4) Conclusão

Conclua o trabalho realizado ressaltando as Limitações do mesmo, Lições Aprendidas durante o desenvolvimento e Trabalhos Futuros que podem ser realizados.

4.1) Contribuições e Limitações

- É importante que fique clara a abrangência da validade dos resultados, mencionando explicitamente as limitações do trabalho. Dependendo do tamanho do texto referentes somente às Contribuições ou às Limitações você pode separar essas duas seções.

4.2) Lições Aprendidas

- Nesta seção deve-se apresentar as facilidades e dificuldades encontradas no decorrer da pesquisa e como isso pode ter interferido positiva ou negativamente no trabalho.

4.3) Trabalhos Futuros

- Apresentar trabalhos que podem ser realizados com base na pesquisa em questão, os quais podem corresponder à continuidade do tema abordado ou a um novo item de pesquisa identificado a partir desse trabalho

Bônus

- Figuras

A Figura 2.1 serve como exemplo de inserção de uma figura no texto. Todo figura deve ser posta no texto após ser citada no mesmo, assim, não insira figuras soltas e desconexas.

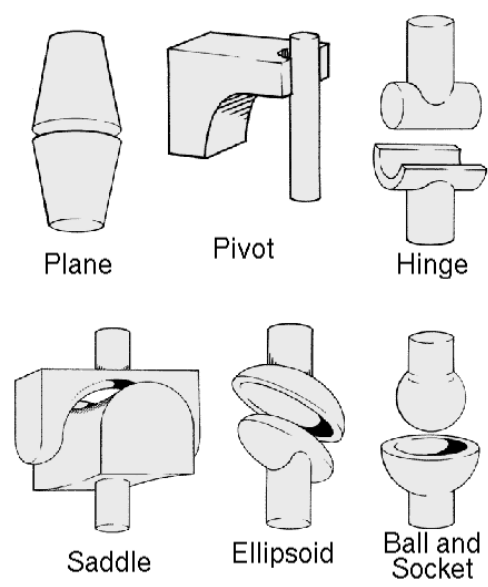


Figura 2.1 - Exemplo de Figura e Legenda. O 2 indica o capítulo em que a figura se encontra o 1 indica que é a primeira figura do capítulo. Utilize Referência Cruzada para facilitar sua vida

- Tabelas

O mesmo vale para as tabelas, abaixo temos a Tabela 2.1 como exemplo de tabela. Note que as legendas das tabelas devem ficar acima das mesmas, diferente da legenda das figuras.

Tabela 2.1 - Exemplo de Tabela e legenda.

Exemplo de Tabela	
Valor 1	Valor 2

REFERÊNCIAS

ABDELNABI, Z. *et al.* Comparing code reading techniques applied to object-oriented software frameworks with regard to effectiveness and defect detection rate. In: INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING - ISESE Washington, USA, 2004. **Proceedings...** Washington: IEEE Computer Society 2004. p. 239-248.

ACKERMAN, A.; BUCHWALD, L.; LEWSKI, F. Software inspections: An effective verification process. **IEEE Software**, v. 6, n. 3, p. 31-37, maio 1989.

ALMEIDA, J. R. D. *et al.* Best practices in code inspection for safety-critical software. **IEEE Software**, v. 20, n. 3, p. 56-63, maio 2003.

AMARAL, E. A. G. G. D.; TRAVASSOS, G. Em busca de uma abordagem para empacotamento de experimentos em engenharia de software. CARVALHO, A. C., *et al* (Ed.) In: JORNADA IBERO-AMERICANA DE ENGENHARIA DE SOFTWARE E ENGENHARIA DO CONHECIMENTO - JIISIC. Salvador, Brasil, 2002. **Anais...** Salvador: JIISIC 2002. p. 75-84.

ABDELNABI, Z. *et al.* Comparing code reading techniques applied to object-oriented software frameworks with regard to effectiveness and defect detection rate. In: INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING - ISESE Washington, USA, 2004. **Proceedings...** Washington: IEEE Computer Society 2004. p. 239-248.

ACKERMAN, A.; BUCHWALD, L.; LEWSKI, F. Software inspections: An effective verification process. **IEEE Software**, v. 6, n. 3, p. 31-37, maio 1989.

ALMEIDA, J. R. D. *et al.* Best practices in code inspection for safety-critical software. **IEEE Software**, v. 20, n. 3, p. 56-63, maio 2003.

AMARAL, E. A. G. G. D.; TRAVASSOS, G. Em busca de uma abordagem para empacotamento de experimentos em engenharia de software. CARVALHO, A. C., *et al* (Ed.) In: JORNADA IBERO-AMERICANA DE ENGENHARIA DE SOFTWARE E ENGENHARIA DO CONHECIMENTO - JIISIC. Salvador, Brasil, 2002. **Anais...** Salvador: JIISIC 2002. p. 75-84.

ABDELNABI, Z. *et al.* Comparing code reading techniques applied to object-oriented software frameworks with regard to effectiveness and defect detection rate. In: INTERNATIONAL SYMPOSIUM ON EMPIRICAL SOFTWARE ENGINEERING - ISESE Washington, USA, 2004. **Proceedings...** Washington: IEEE Computer Society 2004. p. 239-248.

ACKERMAN, A.; BUCHWALD, L.; LEWSKI, F. Software inspections: An effective verification process. **IEEE Software**, v. 6, n. 3, p. 31-37, maio 1989.

ALMEIDA, J. R. D. *et al.* Best practices in code inspection for safety-critical software. **IEEE Software**, v. 20, n. 3, p. 56-63, maio 2003.

AMARAL, E. A. G. G. D.; TRAVASSOS, G. Em busca de uma abordagem para empacotamento de experimentos em engenharia de software. CARVALHO, A. C., *et al* (Ed.) In: JORNADA IBERO-AMERICANA DE ENGENHARIA DE SOFTWARE E ENGENHARIA DO CONHECIMENTO - JIISIC. Salvador, Brasil, 2002. **Anais...** Salvador: JIISIC 2002. p. 75-84.