

ETEP - FACULDADE DE TECNOLOGIA   
DE SÃO JOSÉ DOS CAMPOS

PROPOSTA DE SISTEMA DE CATÁLOGO DIGITAL UTILIZANDO METODOLOGIA BASEADA EM TESTES

João Paulo Gomes dos Santos

Trabalho de Conclusão de Curso de Bacharelado em Engenharia da Computação, orientado pelo Prof. Edizon Basseto Júnior.

ETEP Faculdades

São José dos Campos

2011

ETEP - FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS

PROPOSTA DE SISTEMA DE CATÁLOGO DIGITAL UTILIZANDO METODOLOGIA BASEADA EM TESTES

João Paulo Gomes dos Santos

\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Edizon Basseto Júnior

ETEP Faculdades

São José dos Campos

2011

“Quando estás certo, ninguém se lembra; quando estás errado, ninguém esquece”.

Provérbio irlandês

**PROVÉRBIO OU DEDICATÓRIA OPCIONAL, FEITA PELO AUTOR, EM FORMATO LIVRE, INDICANDO ABAIXO QUAL O TIPO DE PROVÉRBIO OU SE DEDICATÓRIA.**

AGRADECIMENTOS

Nesta parte o autor é livre para fazer seus agradecimentos pessoais, no entanto é aconselhável que não se esqueça de agradecer à Instituição e aos professores que participaram da sua formação, além, obviamente, ao orientador e às contribuições da Banca Examinadora.

RESUMO

O resumo é um texto que sintetiza os aspectos importantes do trabalho ressaltando o objetivo, o método, os resultados e as conclusões do trabalho. Deve ser composto de uma seqüência de frases concisas, afirmativas e não enumeração de tópicos formando um parágrafo único com alinhamento justificado, 150 a 600 palavras e frases com verbo na voz ativa e na terceira pessoa do singular. Recomenda-se evitar o uso de citações de autores, fórmulas, abreviaturas, símbolos, equações, frases negativas etc. Este documento apresenta instruções para a elaboração de monografia para trabalho de conclusão de curso dos cursos de graduação da ETEP FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS. A estrutura básica do modelo para monografia segue orientações das normas para trabalhos acadêmicos da Associação Brasileira de Normas Técnicas e busca a padronização seguindo as seguintes diretrizes: o corpo do texto em uma única coluna; máximo recomendado de 50 páginas, formato tamanho A4 (21 x 29,7 cm), cada qual com margens esquerda e superior iguais a 3 cm e direita e inferior iguais a 2 cm sem molduras e com numeração de página no centro inferior; fonte é *Times New Roman* tamanho 12 pt em todo o documento, espaçamento simples e alinhamento justificado com tabulação de 1,25 cm na primeira linha do parágrafo. As referências devem ser listadas em ordem alfabética no final do trabalho e as tabelas, figuras, fotografias incluídas no trabalho devem ser de boa qualidade e identificadas segundo exemplo do texto.

***Palavras Chave:*** *Palavra 1; Palavra 2, Palavra 3; Palavra 4; Palavra 5*

SUMÁRIO

**Pág.**

LISTA DE FIGURAS 8

LISTA DE TABELAS 9

LISTA DE SÍMBOLOS 10

1 INTRODUÇÃO (estilo Título 1) 11

1.1 OBJETIVO DO TRABALHO (estilo Título 2) 13

1.2 JUSTIFICATIVA (estilo Título 2) 13

2 FUNDAMENTAÇÃO TEÓRICA 14

2.1 TESTES DE SOFTWARE 14

2.1.1 TESTES DE UNIDADE 15

2.1.2 TESTES DE INTEGRAÇÃO 15

2.2 TESTES AUTOMATIZADOS 16

2.3 LINGUAGEM DE PROGRAMAÇÃO RUBY 16

2.4 FRAMEWORK DE DESENVOLVIMENTO RAILS 18

2.5 DESENVOLVIMENTO MOBILE 18

2.5.1 J2ME 18

2.5.2 IPHONE 18

2.5.3 ANDROID 18

2.6 REST 18

2.7 METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE 18

2.7.1 METODOLOGIA EM CASCATA 18

2.7.2 RUP 18

2.7.3 XP 18

2.7.4 LEAN 18

2.7.5 TDD 18

3 METODOLOGIA 20

3.1 ESCOLHA DAS TECNOLOGIAS 20

3.1.1 FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES COM RAILS 20

3.1.2 FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES NO ANDROID 20

3.1.3 FORMATAÇÃO DA PÁGINA E TEXTO (estilo Título 3) 20

3.1.4 FIGURAS E TABELAS (estilo Título 3) 21

3.1.5 EQUAÇÕES E UNIDADES (estilo Título 3) 22

3.1.6 AS REFERÊNCIAS (estilo Título 3) 23

4 RESULTADOS (estilo Título 1) 24

5 CONCLUSÃO (estilo Título 1) 25

APÊNDICE A (estilo APÊNDICE) 29

ANEXO A - ABREVIATURA DOS MESES (estilo ANEXO) 31

LISTA DE FIGURAS

**Pág.**

[Figura 3.1 – Configuração de página em papel A4. 17](#_Toc294361523)

[Figura A.1 - Diagrama de funcionamento. 25](#_Toc294361524)

[Figura A.2 - Como apresentar uma figura longa 26](#_Toc294361525)

[Figura A.3 - Movimento realocar tarefa. 26](#_Toc294361526)

LISTA DE TABELAS

**Pág.**

[Tabela 3.1 – Botões da barra de ferramentas 3](#_Toc238012854)

[Tabela 1- Abreviaturas 3](#_Toc238012855)

LISTA DE SÍMBOLOS

*f* - Vetor das Forças Aplicadas e Giroscópicas

*E* - Energia Cinética

# INTRODUÇÃO (estilo Título 1)

Uma monografia é um instrumento de difusão de conhecimentos direcionado a um público específico e sua estrutura é orientada pela NBR 14724 – Informação e documentação — Trabalhos acadêmicos — Apresentação da Associação Brasileira de Normas Técnicas (ABNT, 2005). As monografias tal como os demais trabalhos científicos e acadêmicos estão estruturados em elementos pré-textuais, textuais e pós-textuais.

Os elementos pré-textuais são aqueles que contem os dados de identificação: título, autoria, agradecimento, resumo, sumario, índices de figuras e de tabelas e lista de símbolos. O título é a expressão indicativa do conteúdo do documento que deve ser apresentado com o mínimo de palavras possível. Na autoria deve ser identificado pelo nome completo de forma direta do autor. O resumo é uma das partes mais consultadas pelos leitores, é por meio dele que o leitor toma a decisão de ler integralmente o trabalho.

A elaboração do resumo deve ser tarefa cuidadosa. É recomendado que no resumo seja ressaltado claramente qual o objetivo, a metodologia, os resultados e as conclusões do trabalho. O resumo deve ser composto de uma seqüência de frases concisas, afirmativas sem enumeração de tópicos. Recomenda-se ainda que seja composto por um único parágrafo, com atenção especial para a primeira frase que deve explicitar claramente o tema principal do documento.

Os elementos textuais que compõem o corpo do trabalho são a introdução, desenvolvimento e conclusão. A introdução é a parte inicial do texto e deve situar o leitor no contexto do tema pesquisado colocando o leitor a par dos antecedentes, das tendências, dos pontos críticos, dos objetivos e dos resultados esperados com o estudo.

A introdução é a primeira parte do trabalho, podendo constituir-se no primeiro capítulo. É na introdução que o autor deve apresentar a visão geral da pesquisa realizada situando o problema no contexto tratado, esclarecer qual é o objetivo estabelecido no projeto de pesquisa e quais as justificativas para sua elaboração. Deve incluir de forma resumida a metodologia utilizada e descrever como o trabalho está organizado, podendo apresentar uma síntese sobre o conteúdo de cada capítulo. Deve ser ressaltado que o objetivo do trabalho deve ficar claro na introdução, pois seu entendimento é de extrema importância. Quando o autor discute o objetivo, ele faz um contrato com o leitor e todo o restante do trabalho deve estar relacionado com este objetivo.

É indicado, neste modelo, que tanto o objetivo do trabalho quanto sua motivação sejam apresentados na forma de subtítulos da introdução.

O desenvolvimento é a parte mais extensa do trabalho, que contém a exposição pormenorizada da pesquisa realizada e dos resultados alcançados. O texto do desenvolvimento do trabalho pode variar em função da abordagem do tema e do método proposto. Para obter mais informação sobre o formato de apresentação de trabalhos acadêmicos, recomenda-se buscar sugestão do professor orientador, consultar os livros de metodologia científica e outros trabalhos acadêmicos disponíveis no acervo da biblioteca da Instituição. Visando organizar a apresentação de seu conteúdo, o desenvolvimento do trabalho deve ser dividido em seções e subseções. O assunto principal será representado por um número que se subdividirá em partes formando assim, as seções primárias, secundárias, terciárias, quaternárias e quinarias seguindo as orientações da NBR 6024- Numeração progressiva das seções de um documento (ABNT, 2003). A adoção deste recurso facilita a sistematização do texto de forma hierarquizada e sua apresentação no Sumário. Na parte referente ao desenvolvimento é esperada a indicação das referencias bibliográficas, já que uma característica essencial do trabalho científico é a necessidade de indicar as fontes de citações apresentadas no texto. As ilustrações, elementos de apoio, têm por objetivo apresentar, graficamente, informações condensadas para explicar e simplificar o entendimento de um texto. Comumente são consideradas como ilustrações as tabelas e equações. As demais ilustrações como fotografias, desenhos, gravuras, diagramas, esquemas, fluxogramas, mapas, organograma são denominadas genericamente e mencionadas no texto como figuras.

**Neste modelo, é recomendado que o desenvolvimento esteja dividido nos seguintes capítulos: Fundamentação Teórica, Metodologia e Resultados.**

A conclusão “é um resumo marcante dos argumentos principais, é síntese interpretativa dos elementos dispersos pelo trabalho e ponto de chegada das deduções lógicas baseadas no desenvolvimento” (CERVO; BERVIAN; SILVA, 2007, p.124). Na conclusão o autor deve apresentar os resultados alcançados e destacar as contribuições obtidas na realização do estudo e pode sugerir que outros trabalhos sobre o tema sejam realizados. Pode ser expressa também como considerações finais e, no caso, evidenciar de forma breve, com clareza e objetividade as deduções obtidas na pesquisa ou levantadas ao longo da discussão do tema, podendo também nesse caso apresentar recomendações e sugestões para trabalhos futuros.

Os elementos pós-textuais são compostos de Referências, Glossários, Apêndices e Anexos. Destinam-se a esclarecer ou complementar o texto, sem, contudo, fazer parte deste. As referências devem seguir as normas da NBR 6023 (ABNT, 2002). Na seção desse modelo são apresentados exemplos de formatação de referências em diversas situações: livros, artigos, documentos oficiais. Os apêndices são textos ou documentos elaborados pelo próprio autor que servem para fundamentar, comprovar ou ilustrar o trabalho. Porém, por serem extensos e para não quebrar a seqüência lógica de exposição do texto, não foram incluídos no corpo do trabalho. Já os anexos são materiais (textos, documentos, figuras, tabelas, formulários, mapas, desenhos etc.) produzidos por outras fontes que não o autor, que servem para fundamentar, comprovar ou ilustrar seu trabalho.

## OBJETIVO DO TRABALHO (estilo Título 2)

Os objetivos devem traduzir as ações que serão realizadas pelo pesquisador para atingir seus fins. Eles podem ser apresentados em dois níveis:

a) objetivo geral que deverá explicitar de modo claro e preciso a grande ação do estudo proposto;

b) objetivos específicos que apresentam ações a serem desenvolvidas para alcançar o objetivo geral.

## JUSTIFICATIVA (estilo Título 2)

A justificativa consiste de uma exposição sobre as razões que motivaram a realização da pesquisa como: o interesse pessoal ou institucional na investigação do tema; a importância teórica e prática do tema da pesquisa; a relevância social; e a oportunidade econômica.

# FUNDAMENTAÇÃO TEÓRICA

## TESTES DE SOFTWARE

Como em qualquer área de desenvolvimento, os testes são essenciais para aumentar a qualidade do produto final, no desenvolvimento de software não é diferente.

A definição de teste de software segundo (MYERS, 2004) diz que teste de software é o processo de executar um software com o intuito de encontrar erros.

Devido à natureza dinâmica dos softwares não é possível garantir que eles fiquem livres de erros (MYERS, 2004), mesmo executando exaustivas baterias de testes.

Segundo (MCGREGOR, 2001) e (GUERRA, 2005), existem vários tipos de teste de software cada um com um propósito específico, alguns dos principais são:

* Teste de unidade: O teste deve ser feito sobre uma unidade do software, ou seja, sobre uma classe ou método. Testes de unidade ajudam a garantir que a unidade a ser testada está desacoplada das outras unidades e coesa. Seu principal objetivo é verificar se a unidade possui o comportamento esperado.
* Teste de Interação: O teste deve ser feito sobre um grupo de classes do software e devem verificar se a troca de mensagens entre as classes ocorre corretamente.
* Teste de Componente: O teste deve ser feito sobre um grupo de classes e devem ser verificadas as entradas e as saídas do componente.
* Teste de Integração: Deve ser feito sobre um grupo de componentes.
* Teste de Sistema: Deve ser realizado utilizando o software como um todo, usando a visão do usuário final. Este teste ajuda na verificação dos requisitos do software.
* Teste de Validação: Assim como o teste de sistema, este teste deve utilizar o software como um todo, tendo como única diferença a necessidade de acompanhamento do cliente durante os testes.

A execução dos testes de forma manual é muito custosa para grandes softwares e assim a execução de todos os testes a cada nova versão do software acaba se tornando inviável para softwares de grande porte (GUERRA, 2005).

A qualidade do software diminui à medida que a qualidade dos testes diminui, ou seja, quanto menor a eficácia dos testes do software menor será a qualidade do produto final, já que a garantia de que tudo esteja funcionando corretamente irá diminuir (GUERRA, 2005).

Nos próximos capítulos serão detalhados os principais tipos de teste utilizados neste trabalho.

### TESTES DE UNIDADE

A unidade fundamental de um sistema orientado a objetos é denominada classe, o teste de unidade de uma classe tem como objetivo principal verificar se a implementação da classe corresponde ao que foi especificado para a mesma (MCGREGOR, 2001).

Garantindo que todas as classes estão cobertas por testes de unidade, seguindo suas especificações, haverá uma grande chance de que qualquer problema no sistema desenvolvido foi causado por erros de integração entre as unidades.

Segundo (MCGREGOR, 2001), o tempo gasto com correção de problemas no sistema é reduzido drásticamente, já que as unidade são coesas e testadas corretamente.

### TESTES DE INTEGRAÇÃO

Um sistema orientado a objetos é composto por um conjunto de objetos que colaboram entre si para atingir um objetivo. O modo com que estes objetos interagem entre si determina o resultado do sistema (MCGREGOR, 2001).

Um exemplo seria o caso de um sistema em que todas as unidades tem o comportamento correto quando isoladas, mas quando colocadas para interagirem entre si o objetivo do sistema não é atingido, ou seja, existe um problema de integração entre as unidades (MCGREGOR, 2001).

Segundo (MCGREGOR, 2001) o principal objeto de um teste de integração é garantir que as mensagens enviadas de um objeto para outro sejam executadas de maneira correta.

Antes de realizar testes de integração deve-se garantir que as unidades participantes do teste estão cobertas por testes de unidade.

## TESTES AUTOMATIZADOS

Executar uma bateria de testes manuais a cada iteração do sistema se torna inviável à medida que o software cresce (GUERRA, 2005), com o intuído de contornar este problema surgiu o conceito de automação de testes.

Automatizar o processo de testes aumenta a confiabilidade do software a ser desenvolvido e garante maior agilidade no ciclo de desenvolvimento, já que os testes de regressão podem ser executados a cada iteração do ciclo de desenvolvimento (DUSTIN, 2002).

Com a execução dos testes de regressão de maneira automatizada podemos saber rapidamente se qualquer uma das modificações efetuadas causou algum impacto indesejado em outras partes do software (DUSTIN, 2002).

Recomenda-se que a cada alteração no sistema seja criado um caso teste para cobrir a nova situação criada com a alteração em questão , depois de elaborar o caso de teste é necessário adicioná-lo ao conjunto de testes do sistema em desenvolvimento, fazendo com que o conjunto de testes automatizados esteja sempre atualizado.

## LINGUAGEM DE PROGRAMAÇÃO RUBY

A linguagem de programação Ruby foi criada por Yuki hiro Matsumoto, no Japão, no ano de 1995, desde então vem se tornando uma linguagem robusta o suficiente para ser utilizada em sistemas de qualquer natureza (MATSUMOTO, 2001).

Ruby é uma linguagem de programação totalmente orientada a objetos, já que tudo em Ruby é um objeto, sem exceções.

Muitas linguagens de programação incorporaram aspectos de programação orientada a objetos, mas poucas conseguem ser totalmente orientadas a objetos assim como Ruby (MATSUMOTO, 2001).

Como exemplo pode-se citar a linguagem de programação Java, ela é classificada como uma linguagem de programação orientada a objetos, mas existem representações de tipos primitivos (integer, double, byte, char, etc) em Java, ou seja, esses tipos primitivos não são objetos, sendo assim Java não é uma linguagem totalmente orientada a objetos (SIERRA, 2008). Em Ruby até os inteiros são objetos da classe FixNum.

Segundo (MATSUMOTO, 2001), quando a linguagem Ruby foi desenvolvida o principal foco de seu criador era gerar uma linguagem que pudesse aumentar a produtividade dos desenvolvedores de forma fácil, com base nessa necessidade do criador da linguagem, Ruby adquiriu algumas características, descritas abaixo (MATSUMOTO, 2001):

* Programação Interativa: Ruby é uma linguagem de script, ou seja, não é necessário compilar o código. Existe um interpretador para facilitar o desenvolvimento.
* Programação Dinâmica: Praticamente tudo que é feito em Ruby é feito em tempo de execução. Os tipos das variáveis, expressões, classes e definições de métodos são determinados em tempo de execução. Uma característica interessante é a possibilidade de alteração de suas classes em tempo de execução de maneira totalmente dinâmica, garantindo grande flexibilidade no desenvolvimento.
* Sintaxe Familiar: A sintaxe da linguagem Ruby é muito parecida com a sintaxe de linguagens renomadas, como por exemplo, Java, Perl, Python, C/C++, essa característica ajuda na disseminação da linguagem.
* Bibliotecas de classes: Ruby possui uma grande quantidade de bibliotecas que já vem com a distribuição padrão e cobrem um vasto domínio de necessidades, começando pelos tipos básicos (strings, arrays, hashes) e indo até tópicos mais avançados, como programação para recursos de rede e threads. Mesmo possuindo uma grande quantidade de bibliotecas nativas, é possível adicionar novas bibliotecas desenvolvidas por terceiros.
* Portabilidade: Programas escritos em Ruby podem ser rodados em qualquer ambiente computacional que possua um interpretador Ruby, ou seja, é possível criar programas Ruby em um plataforma e migrá-los para outra sem a necessidade de nenhuma modificação.

## PADRÃO ARQUITETURAL MVC

Em 1979, Trygve Reenskaug desenvolveu um padrão arquitetural para desenvolvimento de aplicativos, padrão conhecido como Modelo – Visão – Controlador **(MVC)**. Como o próprio nome sugere esse padrão arquitetural divide a arquitetura das aplicações em 3 camadas básicas (RUBY, THOMAS e HANSSON, 2010):

* Modelo: O modelo é responsável por representar os estados dos objetos da aplicação. O estado dos objetos pode ser considerado transiente, quando o estado é mantido apenas por algumas interações, ou persistente, quando o estado é grava em um mecanismo de persistência, como os banco de dados. O modelo é muito mais que apenas dados, ele assegura algumas regras de negócios relacionadas com os dados representados por ele.
* Visão: A visão é responsável por gerar a interface com o usuário, baseando-se nos dados do modelo.
* Controlador: Os controladores são responsáveis por coordenar o funcionamento da aplicação, eles recebem os eventos gerados pela visão, interagem com os modelos e respondem aos eventos delegando o fluxo para outras visões.

## FRAMEWORK DE DESENVOLVIMENTO RAILS

Rails é um framework para desenvolvimento de aplicações para web escrito em Ruby. Criado por David Heinemeier Hansson, tem como seu principal objetivo facilitar o desenvolvimento de aplicações para web, levando em consideração alguns aspectos que os desenvolvedores precisam conhecer para começar a desenvolver uma aplicação (RUBY, THOMAS e HANSSON, 2010).

Os principais conceitos utilizados na criação do Rails foram:

* Convenção ao invés de configuração: Principal fundamento do framework, para tudo no Rails existe uma convenção, como por exemplo:
  + Estrutura de diretórios da aplicação;
  + Padrão de nomes dos controladores, sempre seguindo o nome dos modelos;
  + Padrão de nomes das visões, sempre seguindo os nomes das ações dos controladores;
  + Padrão de nomes de tabelas do banco de dados, sempre o nome do modelo no plural;

Conhecer e obedecer as convenções é muito importante para aproveitar todas as facilidades do framework.

* Não se repita: Grande facilidade para reuso de código através de plugins. Alguns do plugins mais populares para o desenvolvimento Rails são:
  + Devise: É capaz de criar uma estrutura completa para autenticação de usuários;
  + Will Paginate: Serve para paginar os itens nas listagens;
  + PaperClip: Serve para controlar upload de arquivos;

A estrutura interna do Rails é divida nos seguintes componentes:

* Action Controller: Cuida do gerenciamento dos controles das aplicações. Processa as requisições HTTP, extrai os parâmetros e faz o encaminhamento para a ação desejada. Outros serviços provenientes do Action Controller são:
  + Gerenciamento de sessões HTTP;
  + Renderização de templates;
  + Redirecionamentos.
* Action Dispatch: É responsável pelo roteamento das requisições HTTP.
* Action View: É responsável pela geração das respostas para as requisições feitas, por padrão Rails tem suporte a html, xml e json.
* Action Mailer: É utilizado para gerenciar o envio e recebimento de e-mails.
* Active Record: É a base para os modelos nas aplicações Rails. Algumas de suas características são:
  + Possibilita a independência de uma banco de dados específico;
  + É capaz de fazer buscas avançadas;
  + Possibilita a criação de relacionamentos entre os modelos.
* Active Suport: Classes utilitárias usadas por todo o framework Rails.

## DESENVOLVIMENTO MOBILE

### J2ME

### IPHONE

### ANDROID

## REST

## METODOLOGIAS DE DESENVOLVIMENTO DE SOFTWARE

### METODOLOGIA EM CASCATA

### RUP

### XP

### LEAN

### TDD

A estrutura deste modelo é de monografia e os demais tipos de publicações acadêmicas como teses, dissertações e projetos de pesquisa seguem, em linhas gerais, a mesma organização.

Para elaborar o trabalho, o autor deverá criar um arquivo .doc e seguir a estrutura, selecionando as formatações para o conteúdo do seu arquivo.

Para o bom uso do modelo, na criação de qualquer tipo de publicação com o Word, seguem três observações fundamentais:

* O trabalho deve ser elaborado com a mesma versão do Word do início ao fim;
* Ao transferir qualquer texto para o estilo deve ser retirada, primeiramente, toda a formatação anterior para colá-lo no trabalho. O método usado para efetuar esta retirada de formatação é copiar o conteúdo do local onde ele se encontra para a área de transferência e trazer este conteúdo para o documento usando Editar/Colar Especial/Texto não formatado;
* Ao abrir o modelo .dot será gerado um novo documento.doc, o sumário e os índices de figuras e tabelas estarão com o número de páginas. Após alterações, para atualizar os números de páginas basta clicar no campo do sumário ou dos índices e então atualizar campo.

Na parte destinada à fundamentação teórica é feita a revisão bibliográfica pormenorizada referente ao tema tratado.

# METODOLOGIA

Neste capítulo, o texto do modelo e todos os seus elementos, desde títulos e subtítulos passando pelas figuras, tabelas e fórmulas até as referências servirão para indicar a formatação que deverá ser utilizada na elaboração do texto da monografia do Trabalho de Conclusão de Curso, TCC.

## ESCOLHA DAS TECNOLOGIAS

### FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES COM RAILS

### FERRAMENTAS PARA AUTOMATIZAÇÃO DE TESTES NO ANDROID

### FORMATAÇÃO DA PÁGINA E TEXTO (estilo Título 3)

A página está formatada com corpo do texto em uma única coluna, tamanho A4 (21 x 29,7 cm), cada qual com margens esquerda e superior iguais a 3 cm e direita e inferior iguais a 2 cm sem molduras e com numeração de página no centro inferior, fonte é *Times New Roman* tamanho 12 pt em todo o documento, espaçamento simples e alinhamento justificado com tabulação de 1,25 na primeira linha do parágrafo. É recomendável que tenha no máximo 50 páginas sem contar os possíveis anexos e apêndices.

A Figura 3.1 mostra a formatação da página utiliza no modelo. Toda figura deve seguir a numeração da seqüência de apresentação da seguinte forma: o primeiro numeral deve corresponder ao capítulo e o segundo para a seqüência de apresentação no texto. Sua referência no texto deve ser sempre com a primeira letra maiúscula e seguida pelo seu número. O parágrafo da figura deve ser o normal centralizado sem recuo e o da legenda da figura deve ser o estilo disponibilizado como FORMATAÇÃO/FIGURA, pois este está com hiperlink para o índice de figuras.



Figura 3.1 – Configuração de página em papel A4.

### FIGURAS E TABELAS (estilo Título 3)

Uma figura é um elemento demonstrativo de síntese que explica ou complementa visualmente o texto. Qualquer que seja seu tipo (quadros, lâminas, plantas, fotografias, gráficos, organogramas, fluxogramas, esquemas, desenhos e outros) deve apresentar sua identificação na parte inferior, precedida da palavra designativa, seguida de seu número de ordem de ocorrência no texto, do respectivo título. As legendas das ilustrações, breves e claras, devem ser inseridas com absoluta proximidade ao trecho a que se referem.

As tabelas são elementos demonstrativos de síntese que apresentam informações tratadas estatisticamente. Elas deverão ter numeração seqüencial em algarismos arábicos precedidos da palavra Tabela. Os títulos das tabelas deverão ser inseridos na parte superior das mesmas, após hífen que separa o título da respectiva identificação tabular, com letra em tamanho menor do que a utilizada no texto. Não se colocará ponto final nos títulos de tabelas. A Tabela 3.1 mostra um exemplo genérico. Após a tabela deve ser deixado um espaço de uma linha para melhor distribuição do texto na pagina.

Tabela 3.1 – Botões da barra de ferramentas

|  |  |
| --- | --- |
|  | Insere símbolos |
|  | Insere equações |
|  | Insere símbolos |
|  | Insere equações |

Tabelas permeadas com o texto devem ser sediadas na altura em que são citadas. Nas tabelas utilizam-se linhas horizontais e verticais para separar o título das colunas no cabeçalho e fechá-las na parte inferior, evitando-se fios verticais para separar colunas e horizontais para separar linhas. Se uma tabela não couber em uma única folha, a parte inferior não será fechada. O título e o cabeçalho serão repetidos na folha seguinte e a linha inferior figurará somente ao final da tabela. As tabelas devem ter significado próprio, dispensando consultas ao texto. Eventuais notas que sejam necessárias à tabela devem ser postas em seu rodapé, logo abaixo da linha de fechamento.

### EQUAÇÕES E UNIDADES (estilo Título 3)

As equações podem ser inseridas através do MICROSOFT EQUATION. A numeração da equação segue o mesmo critério para figuras e tabelas. O primeiro numeral corresponde ao capitulo que está inserida e o segundo numeral corresponde a seqüência de ocorrência no texto. Veja exemplo para a eq. 3.1.

 (3.1)

onde, NC é o nível de cinza do pixel, correspondente ao valor digital normalizado.

A próxima fórmula deve apresentar numeração na seqüência com o primeiro numeral correspondendo ao capítulo e o segundo a seqüência no texto como pode ser vista na eq. 3.2.

 (3.2)

Todos os dados do trabalho, inclusive aqueles em tabelas e figuras, devem estar em unidades do Sistema Internacional (SI). A vírgula deverá ser o separador entre a parte inteira e a parte decimal de números fracionários.

### AS REFERÊNCIAS (estilo Título 3)

As referências deverão ser listadas logo após as considerações finais como título de seção, REFERÊNCIAS No texto, após o trecho citado, as referências deverão ser apresentadas entre parênteses no seguinte padrão: sobrenome do autor em letras maiúsculas e o ano. Exemplos: um autor: (GOMES, 1998); dois autores: (ABRAMOF & MOTA, 2007); três ou mais autores: (ABRAMOF et al., 2007). Caso ultrapasse cinco linhas, a citação deverá ser apresentada em itálico e com recuo. As referências bibliográficas devem ser apresentadas em ordem alfabética e de acordo com a norma da ABNT - NBR 6023 (ABNT, 2002). As referências devem ser apresentadas em ordem alfabética. Nesse modelo, a seção de REFERÊNCIAS é mostrada na forma de apresentação de referências no caso de normas ou manuais, livros, capítulo de livro, artigo em periódico, monografia, dissertação, tese, artigo de jornal, trabalho em evento, referências disponíveis na internet e também quando existe repetição de uma referência. A estilo de formatação disponível para referências bibliográficas é RefBib. Deixe uma linha de espaço entre uma referência e a subseqüente para melhor distribuição na página.

# RESULTADOS (estilo Título 1)

Esta é a parte mais importante do trabalho durante a apresentação. Nesta seção o estudante deve discutir os resultados obtidos. É aqui que devem ser feitas comparações entre os resultados esperados e os previstos pela teoria e/ou a partir de outras experiências conhecidas. Anomalias e discrepâncias devem ser exploradas e explicadas em termos físicos e matemáticos. As explicações devem se basear nos gráficos e nas tabelas apresentadas nas seções anteriores. Por fim, é nessa seção que os resultados serão resumidos e discutidos. Maiores detalhes devem ser colocados em apêndice.

# CONCLUSÃO (estilo Título 1)

Parte final do texto, na qual se apresentam as conclusões correspondentes aos objetivos ou hipóteses. Na conclusão, podem-se incluir também recomendações, sugerindo futuros desenvolvimentos sobre o tema. O(s) autor (es) devem manifestar seu ponto de vista sobre os resultados obtidos e sobre o alcance deles. Não se permite a inclusão de dados novos nessa parte.

REFERÊNCIAS (estilo REFERÊNCIA)

Exemplo no caso de Normas

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 14724:** Informação e documentação — Trabalhos acadêmicos — Apresentação. Rio de Janeiro, 2005. 9.p.

\_\_\_\_\_\_\_\_\_\_\_ **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro, 2002. 24 p.

\_\_\_\_\_\_\_\_\_\_\_ **NBR 10520:** e documentação: citações em documentos. Rio de Janeiro, 2002. 7 p.

\_\_\_\_\_\_\_\_\_\_\_ **NBR 6024**: numeração progressiva das seções de um documento. Rio de Janeiro, 2003. 3 p.

Exemplo no caso de Livro

CERVO, Amado Luis; BERVIAN, Pedro Alcino; SILVA, Roberto. **Metodologia Científica**. 6. ed. São Paulo: Prentice Hall, 2007.

GOMES, L. V. N. **Desenhando:** um panorama dos sistemas gráficos. Santa Maria: Ed.UFSM, 1998.

Exemplo no caso de Capítulo de Livro

WILLIAMS, J. W. Flow measurement. In: ROUSE, H. (org.). **Engineering hydraulic**s. New York: John Wiley & Sons, 1950. p. 229-309.

Exemplo no caso artigo em periódico

ABRAMOF, P. G., MIRANDA, C. R. B., BELOTO, A. F. ; UETA, A.Y., FERREIRA, N. G. An investigation of natural oxidation process on stain-etched nanoporous silicon by micro-Raman spectroscopy. **Applied Surface Science**, v. 253, p. p. 7065-7068, 2007.

Exemplo no caso de monografia, dissertação e tese

MIRANDA, R. B. Filmes de diamante nanocristalino infiltrados em substratos de silício poroso através das técnicas CVD/CVI. 2009. Tese (Doutorado em Engenharia e Tecnologia Espaciais) - Instituto Nacional de Pesquisas Espaciais.

Exemplo no caso de artigo de jornal

MOREIRA, T. Debate sobre software livre chega ao celular. **Valor Econômic**o, São Paulo, 04 out. 2004. p. B4.

Exemplo no caso de trabalho em evento

ABRAMOF, P. G., MOTA A.C. Exame de desempenho dos estudantes: uma porposta de avaliação na educação em engenharia. In: XXXV CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA, Curitiba. Anais do XXXV Congresso Brasileiro de Educação em Engenharia, 2007.

Exemplo de documento disponível na internet

BRASIL, 2002. Conselho Nacional de Educação, Parecer CNE/CES 1362/2001 – **Diretrizes Curriculares Nacionais dos Cursos de Engenharia**. Despacho do Ministro em 22/02/2002, publicado no DOU de 25 de fevereiro de 2002, Seção 1, p 17. Disponível em: <htlm://portal.mec.gov.br/cne/arquivos/pdf/CES 1/2002> Acesso em 16 junho 2008

GLOSSÁRIO (estilo GLOSSARIO)

Área - conjunto de conteúdos (grupos temáticos comuns) que compõem os diferentes campos do saber.

APÊNDICE A (estilo APÊNDICE)

Elemento opcional. O(s) apêndice(s) são identificados por letras maiúsculas consecutivas e pelos respectivos títulos e contem textos explicativos que não fazem parte do texto da monografia mas que foi elaborado pelo autor,.

A resolução das figuras de qualquer publicação deve ser baixa, de forma a serem lidas em qualquer tipo de rede, sem muita demora. Ver exemplos de figuras em Figura A.1, Figura A.2 e Figura A.3.



Figura A. - Diagrama de funcionamento.

Fonte Adaptada de Tourrilhes (2001)



Figura A.2 - Como apresentar uma figura longa



Figura A.3 - Movimento realocar tarefa.

Fonte: Adaptada de Mauri (2003, p. 17).

A mesma notação deve ser utilizada para tabelas apresentadas nos apêndices.

ANEXO A - ABREVIATURA DOS MESES (estilo ANEXO)

Nos anexos são apresentados textos, mapas tabelas ou figuras que não foram criados pelo autor. São elementos opcionais, que são identificados por meio de letras maiúsculas consecutivas, travessão e pelos respectivos títulos. Vinculados ao trabalho para esclarecimento ou documentação, nem sempre da mesma autoria. A numeração de tabelas e figuras leva em consideração a sequência de ocorrência no texto. Veja exemplo da Tabela 1.

Tabela 1- Abreviaturas

|  |  |  |
| --- | --- | --- |
| **Português** | **Espanhol** | **Italiano** |
| janeiro = jan..  fevereiro = fev.  março = mar..  abril = abr.  maio = maio  junho = jun.  julho = jul.  agosto = ago.  setembro = set.  outubro = out.  novembro = nov.  dezembro = dez.  . | enero = ene  febrero = feb  marzo = mar.  abril = abr.  mayo = mayo  junio = jun.  julio = jul.  agosto = ago. septiembre = sep.  octubre = oct.  noviembre =nov. diciembre = dic. | gennaio = gen.  febbraio = feb.  marzo = mar.  aprile = apr.  maggio = mag. giugno = giu.  luglio = lug.  agosto = ago. settembre = set. ottobre = ott. novembre = nov. dicembre = dic. |

|  |  |  |
| --- | --- | --- |
| **Francês** | **Inglês** | **Alemão** |
| janvier = jan.  février = fév.  mars = mars  avril = avr.  mai = mai  juin = juin  juillet = juil.  août = août  septembre = sept. octobre = oct.  novembre = nov.  décembre = déc. | January = Jan.  February = Feb.  March = Mar.  April = Apr.  May = May  June = June  July = July  August = Aug.  September = Sept.  October = Oct.  November = Nov.  December = Dec. | Januar = Jan.  Februar = Feb.  März = März  April = Apr.  Mai = Mai.  Juni = Juni  Juli = Juli  August = Aug.  September = Sept.  Oktober = Okt.  November = Nov.  Dezember = Dez |