Paulo João Rodrigues Neto

**TÍTULO:**

**SUBTÍTULO (SE HOUVER**)

Dissertação/Tese submetido(a) ao Programa de ... da Universidade Federal de Santa Catarina para a obtenção do Grau de ... em ....

Orientador: Prof. Dr.

Coorientador (se houver): Prof. Dr.

Cidade

Ano

Ficha de identificação da obra elaborada pelo autor

através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

|  |
| --- |
|  |
| A ficha de identificação é elaborada pelo próprio autor  Maiores informações em:  <http://portalbu.ufsc.br/ficha> |
|  |
|  |
|  |
|  |
|  |

Paulo João Rodrigues Neto

**TÍTULO: SUBTÍTULO (SE HOUVER)**

Esta Dissertação/Tese foi julgada adequada para obtenção do Título de “....” e aprovada em sua forma final pelo Programa ...

Local, x de xxxxx de xxxx.

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

Prof. xxx, Dr.

Coordenador do Curso

**Banca Examinadora:**

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

Prof.ª xxxx, Dr.ª

Orientadora

Universidade xxxx

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

Prof.ª xxxx, Dr.ª

Corientadora

Universidade xxxx

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

Prof. xxxx, Dr.

Universidade xxxxxx

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

AGRADECIMENTOS

Inserir os agradecimentos aos colaboradores à execução do trabalho.

Xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.

Texto da Epígrafe. Citação relativa ao tema do trabalho. É opcional. A epígrafe pode também aparecer na abertura de cada seção ou capítulo.

(Autor da epígrafe, ano)

RESUMO

O texto do resumo deve ser digitado, em um único bloco, sem espaço de parágrafo. O resumo deve ser significativo, composto de uma sequência de frases concisas, afirmativas, e não de uma enumeração de tópicos. Não deve conter citações. Deve-se usar o verbo na voz ativa. Abaixo do resumo, deve-se informar as palavras-chave (palavras ou expressões significativas retiradas do texto) ou termos retirados de thesaurus da área.

**Palavras-chave:** Palavra-chave 1. Palavra-chave 2. Palavra-chave 3.

ABSTRACT

Resumo traduzido para outros idiomas, neste caso, inglês. Segue o formato do resumo feito na língua vernácula. As palavras-chave traduzidas, versão em língua estrangeira, são colocadas abaixo do texto precedidas pela expressão “Keywords”, separadas por ponto.

**Keywords:** Keyword 1. Keyword 2. Keyword 3.

LISTA DE FIGURAS

[Figura 1 - Elementos do trabalho acadêmico 29](#_Toc449547046)

LISTA DE QUADROS

[Quadro 1 - Formatação do texto 30](#_Toc447824501)

LISTA DE TABELAS

[Tabela 1 - Médias concentrações urbanas 2010-2011 31](#_Toc449547065)

LISTA DE ABREVIATURAS E SIGLAS

ABNT – Associação Brasileira de Normas Técnicas

IBGE – Instituto Brasileiro de Geografia e Estatística

LISTA DE SÍMBOLOS

Yin Yang

Yin Yang



Estrela de Davi em círculo

SUMÁRIO

[1 INTRODUÇÃO 27](#_Toc476647169)

[1.1 RECOMENDAÇÕES DE USO 27](#_Toc476647170)

[1.2 OBJETIVOS 27](#_Toc476647171)

[1.2.1 Objetivo geral 27](#_Toc476647172)

[1.2.2 Objetivos específicos 27](#_Toc476647173)

[2 DESENVOLVIMENTO 29](#_Toc476647174)

[2.1 EXPOSIÇÃO DO TEMA OU MATÉRIA 29](#_Toc476647175)

[2.1.1 Formatação do texto 29](#_Toc476647176)

[2.1.1.1 As ilustrações 31](#_Toc476647177)

[2.1.1.2 Equações e fórmulas 31](#_Toc476647178)

[2.1.1.2.1 Exemplo tabela. 31](#_Toc476647179)

[3 SEÇÃO 33](#_Toc476647180)

[4 CONCLUSÃO 35](#_Toc476647181)

[REFERÊNCIAS 37](#_Toc476647182)

[APÊNDICE A – Descrição 39](#_Toc476647183)

[ANEXO A – Descrição 41](#_Toc476647184)

# INTRODUÇÃO

Novos produtos baseados em *software* devem superar uma preocupante estatística. Aproximadamente 70% dos produtos de *software* falharão, e um dos pontos críticos responsável por 60% - 80% das falhas em projetos, é uma pobre engenharia da requisitos (KAUR, SENGUPTA, 2011). – a prática de coleta de requisitos de um sistema a partir de usuários, clientes e outros *stakeholders* (CHEMUTURI, 2012). Nesse contexto, uma eficiente coleta de requisitos é necessária para minimizar esses problemas que tem consequências drásticas no desenvolvimento de *software*.

Um caso de uso representa todos os modos de se usar um sistema para atingir os objetivos específicos de um usuário específico (JACOBSON, 2011). O conjunto de todos os casos de uso fornece todas as maneiras úteis de se usar o sistema e ilustra o valor que esse sistema fornecerá. Uma modernização dessa abordagem, chamada de caso de uso 2.0 é uma prática escalável, ágil que usa casos de uso para capturar um conjunto de requisitos e guiar o desenvolvimento incremental do sistema (JACOBSON, 2011).

Jacobson (2011) sugere 6 princípios para a aplicação de casos de uso com sucesso:

1. Simplificar através de estórias;

2. Conhecer o todo;

3. Foco no valor;

4. Construir o sistema em fatias (*slices* em inglês);

5. Entregar o sistema em incrementos; e

6. Adaptar para se enquadrar nas necessidades da equipe.

Atualmente existem muitas ferramentas para plataformas *desktop*, *web* e *mobile* que permitem a criação de casos de uso clássicos.

Contudo, ainda que seja um assunto importante, não existem ferramentas que integrem o uso do caso de uso 2.0 com a gerência de um projeto de desenvolvimento.

Nesse contexto, o objetivo desta pesquisa é desenvolver uma ferramenta que permita a criação de casos de uso 2.0 e a integre com o sistema de gerenciamento de projeto Kanboard. (!!)

**2 Fundamentação teórica**

Este capítulo apresenta os fundamentos dos tópicos essenciais para a compreenção deste trabalho de conclusão de curso. A sessão seguinte detalha os principais tópicos sobre o histórico e uso de casos de uso. Nesse contexto, é especificado uma outra visão sobre casos de uso, denominada *use case 2.0*, descrita na sessão 2.2. A sessão 2.3 Kanban descreve a origem e prática de uma estratégia ágil para gerenciamento de projetos. Concluindo este capítulo, a sessão 2.4. Ferramentas de planejamento, monitoração e controle apresenta um sumário sobre ferramentas que representam a infra-estrutura tecnológica para a prática do Kanban e caso de uso em ambientes modernos de desenvolvimento.

**2.1. Use case**

Requisitos de *software* pode ser definido de duas formas (POHL, 2016): Uma condição ou capacidade de um usuário resolver um problema ou alcançar um objetivo; ou uma condição ou capacidade que precisa ser atingida, ou possuída, por um sistema (ou módulo do sistema), para satisfazer um contrato, padrão, especificação ou outro documento formal.

Nesse contexto, o processo de engenharia de requisitos é composto de 4 atividades fundamentais (POHL, 2016):

1. Licitação: Diferentes técnicas usadas para obter os requisitos de stakeholders e outras fontes para refinar os requisitos.

2. Documentação: Descrição adequada dos requisitos obtidos na atividade de licitação. Diferentes técnicas podem ser usadas para documentar os requisitos, usando linguagem natural ou modelos conceituais.

3. Validação e negociação: Para garantir coerência, os requisitos devem ser validados pelos stakeholders.

4. Gerenciamento de requisitos: Atividades necessárias para estruturar requisitos, prepara-los para que possam ser compreendidos por diferentes pessoas com diferentes responsabilidades, e principalmente manter consistência depois de mudanças para garantir a implementação adequada.

Requisitos de *software* licitados podem ser documentados de diversas formas, como histórias de usuário ou casos de uso por exemplo (WIEGERS, BEATTY, 2013). Nesse cenário, uma técnica importante para a análise de requisitos que tem sido largamente aplicada no desenvolvimento de *software*, é a análise de casos de uso. Um caso de uso é definido como uma lista de ações ou etapas que definem as interações entre um ator – papel que representa um usuário humano ou outro sistema externo que interage com o sistema (OMG, 2017) – e o sistema, para atingir um objetivo específico (JACOBSON et al., 1992). Logo, a análise de casos de uso usa-se de casos de uso para identificar requisitos de um sistema para o desenvolvimento de um sistema de *software*.

Atualmente não há um consenso da comunidade acadêmica e indústria para o formato correto do uso dos casos de uso. Cockburn, 1999 afirma que casos de uso são meramente formas de se escrever requisitos. Consequentemente, diferentes propósitos acarretam em diferentes modelos. Classicamente um caso de uso deve descrever um fluxo principal de atividades de um ator, baseado em um conjunto de premissas, possíveis fluxos alternativos e pós-condições esperadas após a conclusão do caso de uso (COCKBURN, 1999). Uma ferramenta gráfica para ilustrar casos de uso é o diagrama UML de casos de uso. Nesse diagrama, um ator é relacionado a um conjunto de casos de uso (Figura 1).

[incluir figura 1]

Este diagrama pode incluir ainda mais informação sobre o caso de uso com o uso de anotações UML. Por exemplo, a figura 1 mostra a anotação *include*, que implica que o caso de uso X e Y tem um fluxo de atividades em comum, que é representado no caso de uso Z. Da mesma maneira, há uma fluxo de atividades opcional em X, que é representado no caso de uso W.

**2.2. Use case 2.0**

**2.2.1. Histórico**

**2.2.2. Detalhamento**

**2.2.3. Diferenças**

**2.2.4. Slices**

O coração do *use case 2.0* são os casos de uso, as *stories* e os *slices,* que capturam os requerimentos e direcionam o desenvolvimento do sistema.

Um caso de uso sofre várias mudanças de estado a partir de sua identificação inicial até a sua realização pelo sistema:

1)Estabelecer o objetivo do caso de uso.

2)Estruturar a narrativa do caso de uso para identificar os primeiros *slices*.

3)O sistema contém o story mais simples para visualizar o objetivo.

4)O sistema contém stories o suficiente para permitir prover uma solução passível de ser usada.

5)O sistema contém todas as stories do caso de uso.

No entanto, um único caso de uso contém muito trabalho a ser feito e precisa ser dividido em pequenos pedaços chamados *slices*, que são uma ou mais stories selecionadas do caso de uso. Os slices precisam fluir rapidamente, permitindo selecionar quais partes do caso de uso serão entregues e provendo uma unidade adequada para o desenvolvimento e teste.

Um caso de uso sofre várias mudanças de estado a partir de sua identificação inicial até a sua realização pelo sistema:

1)Escopo e extensão das stories esclarecidos.

2)O slice foi preparado, melhorando a narrativa e os casos de teste, definindo caminho para implementar o slice com sucesso.

3)Analizado e entendido o impacto nos componentes do sistema.

4)O sistema de software foi aprimorado para implementar o slice e o slice está pronto para testar.

5)Slice verificado e pronto para ser incluso no release.

Apesar de parecer com o modelo em cascata há uma grande diferença. Na cascata os requerimentos são preparados antes que a análise seja iniciada e toda a análise seja concluída antes do início da implementação e toda a implementação seja concluída antes da verificação iniciar. No use case 2.0 os slices são independentes entre si resultando em paralelismo. Enquanto uma fatia de caso de uso está sendo verificada, outra fatia de caso de uso está sendo implementada e uma terceira está sendo analisada.

Stories é como exploramos os casos de uso com nossos stakeholders, com cada story sendo uma thread do caso de uso.

O modelo de caso de uso visualiza os requisitos como um conjunto de casos de uso, fornecendo uma imagem geral do sistema a ser construído. O modelo define os casos de uso e fornece o contexto para a elaboração dos casos de uso individual.

Os casos de uso são explorados com as stories. Cada caso de uso é descrito em uma narrativa de caso de uso que descreve stories e um conjunto de casos de teste que completam as histórias.

As stories são descritas como um conjunto de fluxos. Estes podem ser complementados com um conjunto de requisitos especiais que irão influenciar as stories, ajudá-lo a atribuir as stories certas às fatias do caso de uso para a implementação e, mais importante, definir os casos de teste corretos.

Propriedades essenciais para um slice são uma lista de suas stories, referências ao caso de uso e aos fluxos que definem as stories, referências aos testes e casos de teste que serão usados para verificar sua conclusão e uma estimativa do trabalho necessário para implementar e testar o slice.

**2.3. Kanban**

No ano de 1991, o termo desenvolvimento agil de \*software\* foi popularizado pelo Manifesto Ágil (http://agilemanifesto.org/). Este manifesto definia formalmente princípios e valores que alteram drasticamente a abordagem clássica e preditiva de desenvolvimento de \*software\*, usado por exemplo no modelo Waterfall (BOEHM, 1988). Devido a natureza inerentemente interativa, incremental e evolucionária, casos de uso se enquadram perfeitamente no contexto de desenvolvimento ágil de \*software\*.

A metodologia de visualização de fluxo de trabalho Kanban foi adaptada da manufatura enxuta utilizado pela compania japonesa Toyota (SUGIMORI, et al., 1977). Um indicador de sucesso nos anos 70 no Japão no contexto da produção baseado em demanda é a habilidade de prever a demanda. Nesse contexto o toyotismo inovou com o kanban era utilizar a demanda real observada (OHNO, 1988).

No contexto de desenvolvimento de \*software\* kanban é uma abordagem que usa um quadro kanban para visualizar tarefas. Dessa forma melhora-se a compreensão do trabalho e fluxo de trabalho. A metodologia kanban também sugere a limitação no progresso de trabalho, assim reduz disperdício de tempo e esforço devido a multitarefas e mudanças de contexto e expõe problemas operacionais e estimula a colaboração para melhorar o sistema (BOEG, 2012). O kanban é embasado em dois conjuntos de princípios, para mudanças de gerenciamento e fornecimentos de servicos, que da enfaze em mudanças evolucionárias e foco no cliente. O método não sugere um número de passos ou procedimentos, contudo estimula mudanças contínuas, incrementais e evolucionárias no sistema. Assim um dos objetivos do kanban é minimizar resistência à mudança e facilita-la (BOEG, 2012).

Para focar no cliente e trabalho que atinge as necessidades do cliente, invés de atividades de desenvolvedores, o kanban define seis práticas gerais:

\* Visualização do trabalho;

\* Limitação do trabalho em andamento;

\* Gerenciamento de fluxo de trabalho;

\* Explicitar politicas;

\* Ciclos de feedback, e

\* Evolução experimental.

A operacionalização desses conceitos é atingido por intermédio de um quadro kanban (tradução livre do termo kanban board). Esse quadro exibe um conjunto de cartões que descrevem uma tarefa a ser resolvida. Esses cartões são organizados no quadro por colunas: uma coluna representa o estado que a tarefa se encontra. Por exemplo, o quadro kanban ilustrado na Figura 1 apresenta x tickets organizados em y colunas.

[incluir figura 1]

Interpretando o quadro, pode-se notar que as tarefas x, y e z estão definidas mas o seu desenvolvimento ainda não começou. As tarefas i e j estão em desenvolvimento, as tarefas k e l estão prontas para revisão, w e s estão sendo revisadas e p, q, r estão prontas para serem empacotadas e entregues para o cliente. Nota-se que na coluna "Trabalho em andamento" há uma limitação de cartões. Assim é possível que uma terceira tarefa seja desenvolvida paralelamente a i e j, contudo não é possível começar uma quarta tarefa enquanto as três primeiras não sejam terminadas.

**2.4. Ferramentas de planejamento, monitoramento e controle.**

Ferramentas como quadro kanban auxilia na visualização do fluxo de trabalho, contudo há a desvantagem do uso do espaço físico. Considerando membros da mesma equipe geograficamente disperços, há uma dificuldade em se manter a paridade entre quadros kanban diferentes. Outra desvantagem de se utilizar meios físicos é a dificuldade de manter um histórico completo (necessário para resgatar pacotes de trabalho já terminados) e backup em caso de sinistros (como uma faxineira insandecida motivada pelo desejo de limpar uma parede coberta de papeis). Com o objetivo de resolver estes problemas, atualmente há uma demanda para ferramentas automatizadas de planejamento, monitoração e controle.

Uma dessas ferramentas é o Kanboard (https://kanboard.net/). O foco em simplicidade e minimalismo pode dar uma aparência amadora para esta ferramenta web, porém suas funcionalidades revelam um sistema poderoso e estável. A última versão (até esta data) da ferramenta, 1.0.46 permite a visualização objetiva das tarefas, facilidade em arrastar tarefas entre colunas, busca, diferentes formas de visualização, ações automáticas, gráficos de Gantt, relatórios de produtividade além de integrações com várias ferramentas, backend de autenticação múltipla e internacionalização para 26 idiomas.

A Figura 2 apresenta o mesmo quadro kanban da Figura 1, porém usando o Kanboard. Nota-se que não há perda de informações; pelo contrário, o uso do Kanboard permite que seja visualizado várias informações como prioridade, complexidade, prazo com mais clareza.

## RECOMENDAÇÕES DE USO

Este *template* foi elaborado no Word 10. Para gerar o sumário automático de acordo com a norma NBR 6027/2012 utilize a sequência abaixo para diferenciação gráfica nas divisões de seção e subseção.

**1 SEÇÃO PRIMÁRIA**

1.1SEÇÃO SECUNDÁRIA

**1.1.1 Seção terciária**

1.1.1.1 Seção quaternária

*1.1.1.1 Seção quinária*

1. Seção primária, use estilo **título 1**
2. Seção secundária, use estilo **título 2**
3. Seção terciária, use estilo **título 3**
4. Seção quaternária, use estilo **título 4**
5. Seção quinaria, use estilo **título 5**
6. Referência, apêndice e anexo, use estilo **título 6**
7. Para citação com mais de três linhas use estilo **citação**

## OBJETIVOS

Descrição...

### Objetivo geral

Descrição...

### Objetivos específicos

Descrição...

# DESENVOLVIMENTO

## EXPOSIÇÃO DO TEMA OU MATÉRIA

É a parte principal e mais extensa do trabalho. Deve apresentar a fundamentação teórica, a metodologia, os resultados e a discussão. Divide-se em seções e subseções conforme a NBR 6024 (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2012). Quanto à sua estrutura, segue as recomendações da norma NBR 14724 de 2011 para preparação de trabalhos acadêmicos (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011). Quanto à formatação, segue o modelo adotado pela UFSC, em formato A5.

Figura 1 - Elementos do trabalho acadêmico.



Fonte: Universidade Federal do Paraná (1996).

### Formatação do texto

No que diz respeito à estrutura do trabalho, o modelo para dissertações e teses adotado pela UFSC segue a NBR 14724 (2011). Porém, em relação à formatação, a UFSC adotou o tamanho A5, que corresponde à metade do A4. Por essa razão, foi necessário fazer uma adequação no tamanho da fonte, espaçamento entrelinhas, margens etc., conforme exposto a seguir.

1. O texto deve ser justificado, digitado em cor preta, podendo utilizar outras cores somente para as ilustrações.
2. Deve-se utilizar papel branco.
3. Os elementos pré-textuais devem iniciar no anverso da folha, com exceção da ficha catalográfica.
4. Os elementos textuais e pós-textuais devem ser digitados no anverso e verso das folhas, com espaçamento simples (1).

Quadro 1 - Formatação do texto.

|  |  |
| --- | --- |
| Cor | Branco |
| Formato do papel | A5 |
| Gramatura | 75 |
| Impressão | Frente e verso |
| Margens | Espelhadas: Superior 2, Inferior: 1,5, Interna 2,5 e Externa: 1,5. |
| Cabeçalho | 0,7 |
| Rodapé | 0,7 |
| Paginação | Externa |
| Alinhamento vertical | Superior |
| Alinhamento do texto | Justificado |
| Alinhamento das referências | À esquerda |
| Fonte sugerida | Times New Roman |
| Tamanho da fonte | 10,5 para o texto, incluindo os títulos das seções e subseções e referências. Para as citações com mais de três linhas, as legendas das ilustrações e tabelas, usar fonte 9,5. |
| Espaçamento entre linhas para o texto e referências | Um (1) simples |
| Espaçamento entre parágrafos | Anterior 0,0; Posterior 0,0 |
| Numeração da seção | O indicativo numérico, em algarismo arábico, de uma seção precede seu título, alinhado à esquerda.  As seções primárias devem começar sempre em páginas ímpares. Deixar um espaço (simples) entre o título da seção e o texto e entre o texto e o título da subseção.  Títulos sem indicativo numérico: errata, agradecimentos, listas de ilustrações, tabelas etc., resumo, sumário, referências, apêndices e anexos. |

Fonte: Universidade Federal de Santa Catarina (2011).

#### As ilustrações

Independentemente do tipo de ilustração (quadro, desenho, figura, fotografia, mapa, entre outros), sua identificação aparece na parte superior, precedida da palavra designativa.

A indicação da fonte consultada deve aparecer na parte inferior, elemento obrigatório mesmo que seja produção do próprio autor. A ilustração deve ser citada no texto e inserida o mais próximo possível do texto a que se refere. (ASSOCIAÇÃO BRASILEIRA DE NORMAS TECNICAS, 2011, p. 11).

#### Equações e fórmulas

As equações e fórmulas devem ser destacadas no texto para facilitar a leitura. Para numerá-las, deve-se usar algarismos arábicos entre parênteses e alinhados à direita. Pode-se usar uma entrelinha maior do que a usada no texto (ASSOCIAÇÃO BRASILEIRA DE NORMAS TECNICAS, 2011).

Exemplo

X2 + Y2 = Z2 (1)

(X2 + Y2) = n (2)

##### Exemplo tabela.

De acordo com Instituto Brasileiro de Geografia e Estatística (1993), tabela é uma forma não discursiva de apresentar informações onde os números representam a informação central.

Tabela 1 - Médias concentrações urbanas 2010-2011

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Média concentração urbana** | **População** | | | **Produto Interno Bruto – PIB (bilhões R$)1, 3** | **Número de empresas2, 3** | | **Número de unidades locais2, 3** |
| **Nome** | **Total** | **No Brasil1** |  | |  |  | |
|  |  |  |  | |  |  | |
| Ji-Paraná (RO) | 116 610 | 116 610 | 1,686 | | 2 734 | 3 082 | |
| Parintins (AM) | 102 033 | 102 033 | 0,675 | | 634 | 683 | |
| Boa Vista (RR) | 298 215 | 298 215 | 4,823 | | 4 852 | 5 187 | |
| Abaetetuba (PA) | 141 100 | 141 100 | 0,534 | | 846 | 895 | |
| Bragança (PA) | 113 227 | 113 227 | 0,452 | | 654 | 686 | |

Fonte: IBGE (2010).

# SEÇÃO

Este *template* tem algumas seções criadas na tentativa de facilitar o seu uso. No entanto, não há um limite máximo ou mínimo de seção a ser utilizado no trabalho. Cabe a cada autor definir a quantidade que melhor atenda às suas necessidades. Lembramos que o início de cada seção começa no anverso da folha.

# CONCLUSÃO

As conclusões devem responder às questões da pesquisa, em relação aos objetivos e às hipóteses. Devem ser breves, podendo apresentar recomendações e sugestões para trabalhos futuros.

###### REFERÊNCIAS

Conjunto padronizado de elementos descritivos retirados de um documento, que permite sua identificação individual (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 2011).

ALVES, Maria Bernardete Martins; ARRUDA, Susana Margareth. **Como fazer referências**: bibliográficas, eletrônicas e demais formas de documento. Florianópolis: Universidade Federal de Santa Catarina, Biblioteca Universitária, c2001. Disponível em: <http://www.bu.ufsc.br/design/framerefer.php>. Acesso em: 11 abr. 2013.

==================================================

KAUR, Rupinder; SENGUPTA, Jyotsna. Software Process Models and Analysis on Failure of Software Development Projects. **International Journal Of Scientific & Engineering Research.** [s.i.], p. 1-4. fev. 2011. (KAUR, SENGUPTA, 2011)

Chemuturi, Murali. Requirements engineering and management for *software* development projects. **Springer Science & Business Media**, 2012

ARRUMAR:

# Citações

\* Jacobson Ivar, Christerson Magnus, Jonsson Patrik, Övergaard Gunnar, Object-Oriented \*software\* Engineering - A Use Case Driven Approach, Addison-Wesley, 1992.

\* OMG Unified Modeling Language (OMG UML), Superstructure, V2.1.2, pp. 586–588

\* Writing Effective Use Cases, 1999d

\* http://agilemanifesto.org/

\* BOEHM, Barry W.. . A spiral model of \*software\* development and enhancement. Computer, v. 21, n. 5, p. 61-72, 1988.

\* SUGIMORI, Y. et al. Toyota production system and kanban system materialization of just-in-time and respect-for-human system. The International Journal of Production Research, v. 15, n. 6, p. 553-564, 1977.

\* OHNO, Taiichi. Toyota production system: beyond large-scale production. crc Press, 1988.

\* BOEG, Jesper. Priming Kanban. InfoQ/Trifork,, 2012.

\* POHL, Klaus. Requirements engineering fundamentals: a study guide for the certified professional for requirements engineering exam-foundation level-IREB compliant. Rocky Nook, Inc., 2016. (https://books.google.com.br/books?id=1VsUDgAAQBAJ&lpg=PT38&ots=vy812UJlhe&dq=Requirements%20engineering%20fundamentals%20pohl&lr&hl=pt-BR&pg=PT169#v=onepage&q&f=false)

\* WIEGERS, Karl; BEATTY, Joy. \*software\* requirements. Pearson Education, 2013.

\* https://kanboard.net/

\* http://stevbros.com/blog/80-new-products-fail-70-of-\*software\*-projects-fail-due-to-poor-requirements.html

\* Chemuturi, Murali. Requirements engineering and management for \*software\* development projects. Springer Science & Business Media, 2012.

\* USE-CASE 2.0 The Guide to Succeeding with Use Cases Ivar Jacobson, 2011

==================================================

ASSOCIÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: informação e documentação: citações em documentos: apresentação. Rio de Janeiro, 2002.

­­­­\_\_\_\_\_\_. **NBR 6024**: informação e documentação: numeração progressiva das seções de um documento escrito: apresentação. Rio de Janeiro, 2012.

\_\_\_\_\_\_. **NBR 14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2011.

INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. Normas de apresentação tabular. Rio de Janeiro: [S.n.], 1993. 60 p.

UNIVERSIDADE FEDERAL DE SANTA CATARINA. Biblioteca Universitária. **Trabalho acadêmico: guia fácil para diagramação**: formato A5. Florianópolis, 2009. Disponível em: <http://www.bu.ufsc.br/design/GuiaRapido2012.pdf>. Acesso em: 11 abr. 2013

###### APÊNDICE A – Descrição

Textos elaborados pelo autor, a fim de completar a sua argumentação. Deve ser precedido da palavra APÊNDICE, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas, quando esgotadas as letras do alfabeto.

Planilha 1 – Modelo A.

|  |  |
| --- | --- |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| xxxx | yyyyyyyyyyyyyyy |
| xxxx | yyyyyyyyyyyyyyy |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| xxxx | yyyyyyyyyyyyyyy |
|  | ttttttttttttttttt |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| ttttttttttttt |  |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
|  | gggggggggggggggggg |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |
| rrrrrrrrrrrrrrrrr | eeeeeeeeeeeeeeeee |

Fonte: Elaborada pelo autor (2016).

###### ANEXO A – Descrição

São documentos não elaborados pelo autor, que servem de fundamentação (mapas, leis, estatutos). Deve ser precedido da palavra ANEXO, identificada por letras maiúsculas consecutivas, travessão e pelo respectivo título. Utilizam-se letras maiúsculas dobradas, quando esgotadas as letras do alfabeto.