



**PODER EXECUTIVO**  
**MINISTÉRIO DA EDUCAÇÃO**  
**UNIVERSIDADE FEDERAL DO AMAZONAS**  
**INSTITUTO DE COMPUTAÇÃO**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA**



**MIT - UM CONJUNTO DE TÉCNICAS DE LEITURA PARA  
INSPEÇÃO DE USABILIDADE EM MODELOS DE PROJETO**

**NATASHA MALVEIRA COSTA VALENTIM**

**Manaus, Fevereiro de 2013**

**NATASHA MALVEIRA COSTA VALENTIM**

**MIT - UM CONJUNTO DE TÉCNICAS DE LEITURA PARA  
INSPEÇÃO DE USABILIDADE EM MODELOS DE PROJETO**

Dissertação submetida ao corpo docente do Programa de Pós-Graduação em Informática da Universidade Federal do Amazonas (PPGI-UFAM) como requisito para a obtenção do título de Mestre em Informática.

**Manaus, Fevereiro de 2013**

A Deus,  
que me deu sabedoria ao longo dessa jornada.

## AGRADECIMENTOS

A Deus. O Autor da minha vida. Sou grata ao Senhor!

Ao meu esposo Diego. Quando estudo nas madrugadas, você não reclama. Quando viajo, você não resmunga. Será que toda mulher tem um anjo como esposo ou fui premiada com o último? Eu te amarei para sempre!

A toda a minha família e amigos, que sempre acreditaram em mim. Aos meus avós Lenir e Jorge, que me deram todo o suporte necessário para eu chegar aonde estou. Ao meu pai Jonir, que sempre investiu em meus estudos e sonhou comigo cada passo dessa jornada. À minha mãe Débora, por torcer por mim e por se alegrar com minhas vitórias. A minha tia Jolene, por todo seu amor, incentivo e sua confiança em mim. Aos meus irmãos, Nathália, Jonir Filho e Isaque, aos meus tios Mário e Josianne, aos meus primos, Joanne, Nicolas, Luana e Laís, e à minha madrastra Ivanete, por todo o afeto e auxílio ao longo dessa caminhada. Aos meus sogros, Rosa e Valdemiro, por terem orado para que Deus fosse abrindo as portas na minha vida acadêmica. Ao meu pastor Aroldo e sua família, por me acompanharem espiritualmente e por me levarem pra mais perto de Deus. À família Lelis, pelas inúmeras caronas até a UFAM. À minha amada igreja IBAPE, que sempre me apoiou através das orações.

À professora Tayana Conte, que aceitou me orientar no Mestrado. Muito obrigada por todo o seu conhecimento compartilhado, sua dedicação a este trabalho em todos os momentos, sua confiança e seu apoio constante!

À professora Káthia Marçal, que aceitou me co-orientar quando ainda nem nos conhecíamos pessoalmente.

Aos professores Gleison Santos, Rogério do Nascimento e Káthia Marçal, por aceitarem participar de minha banca.

Ao Grupo de Usabilidade e Engenharia de *Software* da UFAM, pela cooperação e compartilhamento de conhecimento. Aprendi muito em todas as nossas reuniões!

A todos os que participaram da realização dos estudos apresentados neste trabalho.

À Universidade Federal do Amazonas e Instituto de Computação, pelo seu apoio durante todo o mestrado.

À FAPEAM, pelo apoio financeiro ao longo de todo o mestrado.

# SUMÁRIO

<b>LISTA DE FIGURAS .....</b>	<b>VII</b>
<b>LISTA DE TABELAS .....</b>	<b>IX</b>
<b>RESUMO .....</b>	<b>X</b>
<b>ABSTRACT .....</b>	<b>XI</b>
<b>CAPÍTULO 1 – INTRODUÇÃO.....</b>	<b>1</b>
1.1 MOTIVAÇÃO.....	1
1.2 PROBLEMA .....	2
1.3 OBJETIVOS .....	3
1.4 METODOLOGIA.....	5
1.5 ORGANIZAÇÃO.....	6
<b>CAPÍTULO 2 - AVALIAÇÃO DE USABILIDADE DE MODELOS DE PROJETOS DE APLICAÇÕES WEB.....</b>	<b>8</b>
2.1 INTRODUÇÃO.....	8
2.2 MÉTODOS DE AVALIAÇÃO DE USABILIDADE.....	8
2.2.1 <i>Avaliação Heurística</i> .....	10
2.3 AVALIAÇÃO DE USABILIDADE DE APLICAÇÕES WEB.....	12
2.4 TÉCNICAS DE INSPEÇÃO DE MODELOS.....	13
2.4.1 <i>Mapeamento Sistemático sobre Técnicas de Inspeção de Modelos que garantam a Usabilidade de Aplicações Web</i> .....	14
2.4.2 <i>Artigos Selecionados Após a Condução o Mapeamento Sistemático</i> .....	17
2.4.3 <i>Categorização das Técnicas Encontradas e Análise</i> .....	21
2.5 CONSIDERAÇÕES FINAIS.....	25
<b>CAPÍTULO 3 - PROPOSTA INICIAL DO CONJUNTO DE TÉCNICAS MIT .....</b>	<b>26</b>
3.1 INTRODUÇÃO.....	26
3.2 ANÁLISE DA VIABILIDADE DE USO DAS TÉCNICAS .....	27
3.3 DESCRIÇÃO DAS TÉCNICAS SELECIONADAS .....	31
3.3.1 <i>Avaliação de Usabilidade em Casos de Uso</i> .....	31
3.3.2 <i>Avaliação de Usabilidade em Mockups</i> .....	33
3.3.3 <i>Avaliação de Usabilidade em Diagramas de Atividades</i> .....	34
3.4 PROPOSTA DO CONJUNTO DE TÉCNICAS MIT.....	35
3.5 CONSIDERAÇÕES FINAIS.....	44
<b>CAPÍTULO 4 - AVALIAÇÃO DO CONJUNTO DE TÉCNICAS MIT ATRAVÉS DE UM ESTUDO EXPERIMENTAL.....</b>	<b>46</b>
4.1 INTRODUÇÃO.....	46
4.2 ESTUDO DE VIABILIDADE .....	46

4.2.1	<i>Planejamento do Estudo de Viabilidade</i> .....	46
4.2.2	<i>Execução da Inspeção de Usabilidade</i> .....	50
4.2.3	<i>Análise Quantitativa</i> .....	51
4.2.4	<i>Análise Qualitativa</i> .....	58
4.2.5	<i>Ameaças à Validade</i> .....	68
4.3	CONSIDERAÇÕES FINAIS.....	69
<b>CAPÍTULO 5 - CONSTRUÇÃO DA VERSÃO ATUAL DO CONJUNTO DE TÉCNICAS MIT</b> .....		<b>70</b>
5.1	INTRODUÇÃO.....	70
5.2	MELHORIAS NA MIT 1 v1 .....	70
5.3	MELHORIAS NA MIT 2 v1 .....	72
5.4	MELHORIAS NA MIT 3 v1 .....	73
5.5	CONSIDERAÇÕES FINAIS.....	74
<b>CAPÍTULO 6 - CONCLUSÕES E PERSPECTIVAS FUTURAS</b> .....		<b>75</b>
6.1	CONCLUSÕES .....	75
6.2	CONTRIBUIÇÕES.....	76
6.3	PERSPECTIVAS FUTURAS .....	77
6.3.1	<i>Evolução do Mapeamento Sistemático</i> .....	78
6.3.2	<i>Realização de um Novo Estudo de Viabilidade</i> .....	78
6.3.3	<i>Realização de Estudo de Observação</i> .....	78
6.3.4	<i>Realização de Estudo de Caso na Indústria</i> .....	79
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....		<b>80</b>
<b>APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DAS MITS UTILIZADO NO ESTUDO DE VIABILIDADE</b> .....		<b>86</b>
<b>APÊNDICE B – MIT V2</b> .....		<b>88</b>
<b>APÊNDICE C – INSTRUÇÕES</b> .....		<b>96</b>
<b>APÊNDICE D – PLANILHA PARA ANOTAÇÕES DAS DISCREPÂNCIAS</b> .....		<b>97</b>

## LISTA DE FIGURAS

Figura 1 - Visão Geral da Metodologia.....	6
Figura 2 - Visão Geral do Conjunto de Técnicas MIT.....	36
Figura 3 - Itens de verificação da heurística Visibilidade do status do Sistema (MIT 1 v1)	37
Figura 4 - Itens de verificação da heurística Controle e liberdade ao usuário (MIT 2 v1)...	37
Figura 5 - Itens de verificação da heurística Concordância entre o sistema e o mundo real (MIT 3 v1).....	38
Figura 6 – Formação do Conjunto de Técnicas MIT.....	38
Figura 7 – Exemplo de defeito inserido no Caso de Uso.....	47
Figura 8 – Exemplo de defeito inserido no <i>Mockup</i> .....	47
Figura 9 – Exemplo de defeito inserido no Diagrama de Atividades.....	48
Figura 10 – Exemplo da planilha de anotação de discrepâncias.....	50
Figura 11 – Planilha de discrepâncias de um inspetor durante o uso da MIT 1.....	51
Figura 12 – Planilha de discrepâncias de um inspetor durante o uso da MIT 2.....	51
Figura 13 – Planilha de discrepâncias de um inspetor durante o uso da MIT 3.....	51
Figura 14 – Boxplots de eficiência por técnica - MIT 1 X AH.....	53
Figura 15 - Boxplots de eficácia por técnica - MIT 1 X AH.....	54
Figura 16 - Boxplots de eficiência por técnica - MIT 2 X AH.....	55
Figura 17 - Boxplots de eficácia por técnica - MIT 2 X AH.....	56
Figura 18 - Boxplots de eficiência por técnica - MIT 3 X AH.....	57
Figura 19 - Boxplots de eficácia por técnica - MIT 3 X AH.....	58
Figura 20 - Resultado das Questões relacionadas à Percepção sobre a facilidade de uso das MITs. ....	60
Figura 21 - Resultado das Questões relacionadas à Percepção sobre a utilidade das MITs.	61
Figura 22 - Resultado das Questões relacionadas à Facilidade de Compreensão das MITs.	61
Figura 23 - Esquema gráfico da categoria Estrutura da MIT 1.....	63
Figura 24 - Esquema gráfico da categoria Estrutura da MIT 2.....	64
Figura 25 - Esquema gráfico da categoria Estrutura da MIT 3.....	64
Figura 26 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 1.....	65
Figura 27 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 2.....	66
Figura 28 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 3.....	66
Figura 29 - Esquema gráfico da análise de comparação da categoria Estrutura das Técnicas .....	67

Figura 30 - Esquema gráfico da análise de comparação da categoria Facilidade da Aplicação .....	68
Figura 31 - Item de verificação 1AE3 da Versão 2 da MIT 1 .....	70
Figura 32 - Item de verificação 1AD3 da Versão 2 da MIT 1 .....	70
Figura 33 - Alto e Baixo Detalhamento da Versão 2 da MIT 1 .....	71
Figura 34 - Alto e Baixo Detalhamento da Versão 2 da MIT 2.....	72
Figura 35 - Item de verificação 3C3 da Versão 2 da MIT 3 .....	73

## LISTA DE TABELAS

Tabela 1 - Heurísticas de NIELSEN (1994a).....	11
Tabela 2 - Fontes utilizadas.....	15
Tabela 3 - Termos utilizados.....	15
Tabela 4 - Tabela para extração de dados.....	17
Tabela 5 - Totais de artigos retornados, selecionados no 1º filtro e selecionados após o 2º filtro.....	18
Tabela 6 - Técnicas para inspeção de modelos que garantem a usabilidade.....	18
Tabela 7 - Questões Específicas para Caracterização das Técnicas de Inspeção de Modelos.....	21
Tabela 8 - Classificação das Técnicas em relação às Questões Específicas.....	22
Tabela 9 - Análise da Viabilidade de Uso das Técnicas.....	27
Tabela 10 - Heurísticas do Método UCE propostas por HORNBACK et al. (2007).....	32
Tabela 11 - Primeira versão das MITs.....	38
Tabela 12 - Caracterização dos Inspectores.....	49
Tabela 13 - Resultados por Inspetor – MIT 1 X AH.....	52
Tabela 14 - Resultados dos indicadores de Eficiência e Eficácia – MIT 1 X AH.....	52
Tabela 15 - Resultados por Inspetor – MIT 2 X AH.....	54
Tabela 16 - Resultados dos indicadores de Eficiência e Eficácia – MIT 2 X AH.....	55
Tabela 17 - Resultados por Inspetor – MIT 3 X AH.....	56
Tabela 18 - Resultados dos indicadores de Eficiência e Eficácia – MIT 3 X AH.....	57
Tabela 19 - Questões e afirmações do questionário pós-inspeção.....	59

## RESUMO

### MIT - UM CONJUNTO DE TÉCNICAS DE LEITURA PARA INSPEÇÃO DE USABILIDADE EM MODELOS DE PROJETO

Orientadora: Tayana Uchôa Conte

Co-orientadora: Káthia Marçal de Oliveira

Esta dissertação apresenta o mapeamento sistemático realizado sobre Técnicas de Inspeção de Modelos que garantam a Usabilidade e a categorização e análise destas técnicas. Além disso, esta dissertação apresenta um conjunto de técnicas de leitura específicas para inspeção de usabilidade em modelos de projeto de desenvolvimento de *software*, a MIT (*Model Inspection Technique for Usability Evaluation*). Estas técnicas possuem um conjunto de itens de verificação que guiam o inspetor durante a avaliação de usabilidade. O seu propósito é que seja facilmente adotável pela indústria de desenvolvimento de *software*, além da possibilidade de serem usadas de forma independente ou combinada, ou seja, seu uso poderá variar de acordo com o processo de desenvolvimento adotado. Como as MITs tiveram sua construção e avaliação apoiada por experimentação, esta dissertação também aborda sobre o estudo de viabilidade realizado, além da evolução das técnicas.

Palavras-chave: MIT, Usabilidade, Inspeções de Usabilidade, Usabilidade de Modelos de Projeto, Integração de IHC (Interação Homem-Computador) e ES (Engenharia de *Software*), Técnica de Leitura.

## ABSTRACT

### MIT - A SET OF READING TECHNIQUES FOR USABILITY INSPECTION IN DESIGN MODELS

Advisor: Tayana Uchôa Conte

Co-advisor: Káthia Marçal de Oliveira

This thesis presents a systematic mapping about techniques for models inspection to ensure the usability and the categorization and analysis these techniques. Moreover, this thesis presents a set of reading techniques specific for usability inspection in design models of software development, MIT (Model Inspection Technique for Usability Evaluation). These techniques have a set of check items that guide the inspector during the usability evaluation. Their purpose is be easily adoptable by software development industry, besides the possibility of be used independently or combined, i.e., their use might vary according with the procedure of development adopted. As the MITs had its construction and evaluation supported by experimentation, this thesis also discusses about the study on academy, as well as evolution of the techniques.

Keywords: MIT, Usability, Usability Inspection, Usability of Design Models, integration of HCI (Human-Computer Interaction) and SE (Software Engineering), Reading Technique.

# CAPÍTULO 1 – INTRODUÇÃO

## 1.1 Motivação

Nos últimos anos a demanda por desenvolvimento de aplicações *Web* vem crescendo sistematicamente e, atualmente, essa categoria de *software* já representa um esforço considerável de desenvolvimento no contexto geral de aplicações de *software* (CONTE *et al.* 2009). Esse crescimento pode ser observado não só do ponto de vista da quantidade de aplicações, mas também no que diz respeito à complexidade, criticalidade e, conseqüentemente, aos riscos para o negócio de diversas organizações fortemente apoiadas por aplicações *Web*. A exploração de conceitos de ubiqüidade e facilidade de acesso a informações e serviços proporcionados pela *Web* vêm permitindo o uso cada vez mais intenso dessas aplicações na sociedade como um todo (KAPPEL *et al.* 2006b), apoiando as relações entre empresas e indivíduos em atividades do dia-a-dia relacionadas ao trabalho ou lazer (CONTE *et al.* 2009).

Porém, de início, é importante apresentar a definição de aplicações *Web*. Como não há um consenso da comunidade científica em torno desta definição, nesta dissertação, adotou-se a definição adaptada de KAPPEL *et al.* (2006a), apresentada em CONTE (2009):

*“Uma aplicação Web é um sistema de software baseado em tecnologias e padrões do World Wide Web Consortium (W3C) que provê recursos específicos de Web, como conteúdo e serviços, através de um cliente Web”.*

Segundo OFFUTT (2002), os três critérios de qualidade que dirigem o processo de desenvolvimento em organizações que produzem aplicações *Web* são: confiabilidade, segurança e usabilidade.

O trabalho aqui apresentado foca no critério usabilidade, que é um componente importante da qualidade de *software* (JURISTO *et al.* 2004) e tem sido cada vez mais reconhecida como um atributo de qualidade que deve ser planejado durante o desenvolvimento de *software*. De acordo com a norma ISO 9240-11 (ISO/IEC 1998), a usabilidade é definida como a “medida que um produto pode ser usado por um usuário específico para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico”. No caso específico de aplicações *Web*, a usabilidade possibilita uma vantagem competitiva, já que a facilidade ou a dificuldade do usuário ao interagir com esta implicará no sucesso ou fracasso destas aplicações (DONAHUE 2001).

Uma comunidade científica que tem investido esforços para garantir a usabilidade dessas aplicações *Web* é a comunidade de IHC (Interação Humano-Computador). Outra comunidade, chamada ES (Engenharia de *Software*) tem inserido em seus projetos quesitos ligados a usabilidade. Porém, estas comunidades evoluíram separadamente e cada uma desenvolveu seus próprios métodos para atender às necessidades dos seus clientes e usuários de *software* (JURISTO *et al.* 2007a).

Nos últimos vinte anos, as fronteiras entre IHC e ES têm diminuído e as tentativas de minimizar a distância entre elas têm aumentado (JURISTO *et al.* 2007a). SEFFAH *et al.* (2001) propõem que os engenheiros de *software* e engenheiros de usabilidade aprendam uns com os outros para facilitar e incentivar a convergência das práticas em ambas as comunidades. Portanto é de grande importância promover a mútua compreensão das atividades e responsabilidades das duas comunidades na construção de *software* com boa usabilidade, garantindo, assim, que os problemas de usabilidade serão adequadamente tratados no projeto de *software* durante todo o ciclo de desenvolvimento (JURISTO *et al.* 2007a).

Várias empresas de desenvolvimento de *software* estão começando a integrar práticas de IHC em seus processos de Engenharia de *Software* (FERRÉ *et al.* 2004). Um indício da importância da integração de questões de usabilidade com o processo de desenvolvimento está no Processo de Usabilidade criado na primeira alteração à Norma ISO/IEC 12207 para Processos de Ciclo de Vida do *software* (ISO/IEC 2002), onde a primeira atividade deste processo trata da especificação de como atividades centradas no usuário se encaixam em todo o processo do ciclo de vida do *software* e seleciona métodos e técnicas de usabilidade. Assim esta integração não deve ser uma área de interesse de alguns especialistas (FERRÉ *et al.* 2005), já que esta é essencial para desenvolvimento de Aplicações *Web* devido à importância da usabilidade nestes tipos de aplicação.

## 1.2 Problema

Algumas organizações já fazem uso de técnicas de garantia da qualidade, como revisões e inspeções para avaliar as aplicações *Web*, porém, segundo ABRAHÃO *et al.* (2003), a qualidade das aplicações *Web* tem sido frequentemente avaliada de maneira *ad hoc*. Para solucionar esse problema, faz-se necessário: (1) definir novas técnicas relacionadas à qualidade de processos e produtos em Engenharia de Aplicações *Web*; e (2) avaliar e disseminar a tecnologia desenvolvida, transferindo-a de modo seguro para as organizações que desenvolvem aplicações *Web* (CONTE *et al.* 2009).

Algumas técnicas de avaliação de usabilidade têm sido utilizadas nas últimas fases do desenvolvimento de *software* (CONTE *et al.* 2009; MORAGA *et al.* 2007; OLSINA e ROSSI 2002), com a avaliação da aplicação final ou mesmo de uma versão preliminar da interface final. No entanto, estudos mostram que se inspeções forem realizadas durante as fases iniciais do desenvolvimento de uma aplicação, elas auxiliam a revelar problemas que possam ser corrigidos a um custo menor (TRAVASSOS *et al.* 1999).

Segundo BELGAMO *et al.* (2005), todas as fases que compõem o processo de desenvolvimento de *software* têm sua importância e contribuem de forma clara e definida para o processo como um todo. Essas fases produzem artefatos intermediários (modelos) que, conforme vão evoluindo, quer seja por um modelo de projeto sequencial ou por um modelo de projeto iterativo e incremental, acabam se transformando no produto final – o *software*. Tanto os artefatos quanto o *software* devem ser avaliados em relação à sua qualidade. Para JURISTO *et al.* (2007b), as avaliações de usabilidade devem também ser realizadas nos primeiros estágios do processo de desenvolvimento *Web*, a fim de garantir uma boa experiência de interação entre o usuário e a interface.

Técnicas têm sido propostas para garantir um bom nível de usabilidade das aplicações *Web* nas fases iniciais do processo de desenvolvimento, a chamada “*Early Usability*” (FERNANDEZ 2009a; LUNA *et al.* 2010; MOLINA e TOVAL 2009; PANACH *et al.* 2007). Uma parte destas técnicas objetiva a garantia da usabilidade através da inspeção de modelos usados no projeto das aplicações. Existem técnicas de inspeção aplicáveis a diferentes modelos de projeto, desde diagramas padronizados da UML<sup>1</sup> (*Unified Modeling Language*) até modelos propostos por um método específico.

Porém, estas técnicas focam em uma parte restrita de modelos, limitando assim uma abrangência maior na busca por problemas de usabilidade nas fases iniciais do processo de desenvolvimento de *software*. Com isso, o problema abordado neste trabalho está relacionado com a qualidade das aplicações *Web* durante o seu desenvolvimento, buscando reduzir o número de defeitos de usabilidade nos modelos de análise e projeto usado nas fases iniciais do desenvolvimento de aplicações *Web*.

### 1.3 Objetivos

O objetivo principal deste trabalho consiste em propor técnicas que apoiem a avaliação de usabilidade de aplicações *Web* nos estágios iniciais de desenvolvimento (análise e projeto), permitindo a identificação de um maior número de problemas de usabilidade

---

<sup>1</sup> <http://www.uml.org/>

durante seu desenvolvimento com um esforço reduzido, se comparado a uma abordagem convencional.

Os objetivos específicos são:

- Identificar características dos métodos de avaliação de usabilidade em modelos de análise e projeto;
- Desenvolver um conjunto de técnicas de inspeção de modelos de análise e projeto para garantir a usabilidade de aplicações *Web*;
- Evoluir as técnicas propostas, a partir dos resultados de estudos experimentais que serão realizados para a avaliação da tecnologia;
- Definir um conjunto de recursos que apoiem a aplicação das técnicas de inspeção de usabilidade em modelos.

O propósito final é que a técnica definida possa ser empregada pelos próprios envolvidos nos projetos de desenvolvimento de *software* durante a fase de análise e projeto. A meta ao utilizar esta técnica é encontrar problemas de usabilidade na aplicação *Web* já através de seus modelos de análise e projeto, contribuindo então para melhorar a usabilidade da aplicação *Web* nas fases iniciais de desenvolvimento.

Para alcançar este propósito a técnica deve ser baseada nos atributos abaixo que, segundo BOLCHINI e GARZOTTO (2007), podem contribuir para a aceitação e a adoção do método de avaliação de usabilidade pela indústria de *software*.

- Ser fácil de aprender e de utilizar (menor tempo para aprender e aplicar a técnica);
- Apresentar bom nível de eficácia (razão entre o número de defeitos detectados e o número total de defeitos existentes);
- Apresentar bom nível de eficiência (razão entre o número de defeitos por tempo de inspeção);
- Oferecer uma boa relação custo-benefício na sua aplicação (custo calculado através da soma dos custos de capacitação, contratação de especialistas e homens-hora empregados na inspeção).

Além disso, a técnica deve:

- Ser aplicada em modelos de análise e projeto;
- Não ter a obrigatoriedade do uso de ferramenta para a utilização da técnica, pois na maioria das vezes as ferramentas não são gratuitas, o que torna o seu uso custoso.

## 1.4 Metodologia

Segundo MENDES (2005), o processo científico dá apoio à construção do conhecimento. Este por sua vez, envolve a utilização de estudos experimentais para testar modelos e hipóteses anteriormente propostos, assegurando que o entendimento atual do campo é correto. É de extrema importância que a pesquisa passe pela fase de experimentação, provando assim que o que se está afirmando é válido. Portanto, a pesquisa aqui apresentada se apoia na experimentação.

SHULL *et al.* (2004) afirmam que estudos experimentais devem ser realizados para melhorar a credibilidade da pesquisa em Engenharia de *Software*, deixando de forma pública a outros pesquisadores o conhecimento usado na execução de um experimento e possibilitando um melhor entendimento e análise do estudo realizado. A meta com isto é construir um corpo de conhecimento baseado em experimentação que identifica as vantagens e os custos das diferentes técnicas e ferramentas de apoio à Engenharia de *Software*.

Há uma classificação para estudos experimentais apresentada por TRAVASSOS e BARROS (2003):

- Estudos *in vitro*: estudos realizados em um ambiente controlado, com a participação de pessoas que atuam como representantes da população de interesse.
- Estudos *in vivo*: estudos que envolvem pessoas em seu próprio ambiente de trabalho em condições realistas.
- Estudos *in virtuo*: estudos que envolvem a interação entre participantes reais e um modelo computacional da realidade.
- Estudos *in silico*: estudos onde tanto os participantes quanto o mundo real são descritos como modelos computacionais.

Para apoiar a definição e o aprimoramento da técnica de inspeção de modelos de análise e projeto que garantam a usabilidade de aplicações *Web* desenvolvida neste trabalho, a metodologia adotada nesta pesquisa é baseada na realização de estudos experimentais. Inicialmente foi realizado um levantamento das técnicas de inspeção de modelos de aplicações *Web* citados na literatura técnica, com o objetivo de ajudar a identificar características para propor a nova técnica. Com isso, foi feito um mapeamento sistemático da literatura (KITCHENHAM 2004) com busca em artigos de periódicos e conferências indexados nas bibliotecas IEEE, ACM e Scopus e nos anais de 2010 e 2011 do Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC). O processo seguido neste mapeamento, assim como todas as referências encontradas nas bibliotecas digitais

selecionadas, está descrito detalhadamente em dois Relatórios Técnicos (VALENTIM e CONTE 2011) (VALENTIM *et al.* 2012a) e também será abordado no Capítulo 2.

O mapeamento sistemático realizado forneceu subsídios para a definição da primeira versão do conjunto proposto de técnicas para inspeção de modelos. Esta versão inicial do conjunto de técnicas é apresentada em um Relatório Técnico (VALENTIM e CONTE 2012) e também será apresentada no Capítulo 3.

Após a definição inicial, foi realizado um estudo *in vitro* (estudo de viabilidade realizado com alunos como participantes). Este estudo possibilitou o acompanhamento do desenvolvimento do conjunto proposto de técnica e teve como objetivo avaliar a viabilidade de uso das técnicas propostas. Após este estudo foi feita uma revisão da versão inicial do conjunto proposto de técnicas (Figura 1).

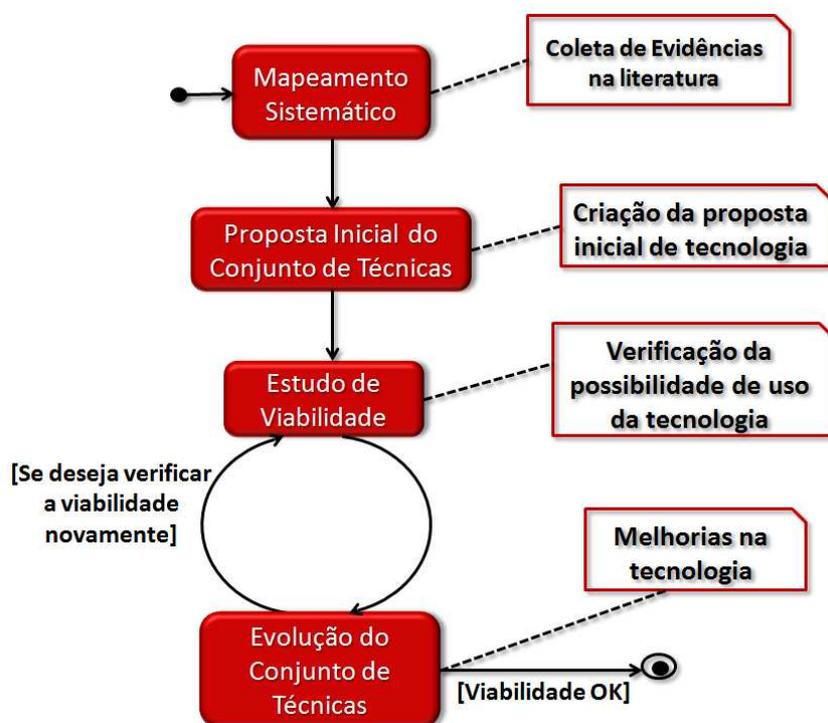


Figura 1 - Visão Geral da Metodologia

## 1.5 Organização

Esta tese está organizada em outros quatro capítulos, além deste primeiro capítulo de introdução, que apresentou a motivação, o problema, os objetivos e a metodologia. A organização do texto deste trabalho segue a estrutura abaixo:

- **Capítulo 2 – Técnicas de Inspeção de Modelos que Garantam a Usabilidade de Aplicações Web:** descreve os tipos de avaliações de usabilidade existentes assim como as específicas para as etapas iniciais do processo de desenvolvimento

de aplicações *Web*. Além disso, descreve os resultados de um estudo secundário (mapeamento sistemático) realizado com o propósito de caracterizar e analisar as técnicas de inspeção de modelos de análise e projeto que garantam a usabilidade de aplicações *Web*.

- **Capítulo 3 – Proposta Inicial do Conjunto de Técnicas MIT:** descreve os métodos de inspeção de usabilidade que foram selecionados como base para esta proposta. Além disso, descreve a proposta inicial de um conjunto de técnicas que garantam a usabilidade de aplicações *Web* através da avaliação dos modelos de análise e projeto e que tem o propósito de ser facilmente adotável pela indústria de *software*.
- **Capítulo 4 – Avaliação do Conjunto de Técnicas MIT através de um Estudo Experimental:** descreve o estudo de viabilidade executado com o propósito de avaliar e aperfeiçoar as técnicas MITs.
- **Capítulo 5 - Construção da Versão Atual do Conjunto de Técnicas MIT:** apresenta as modificações que foram realizadas na técnica após a realização do estudo experimental.
- **Capítulo 6 - Conclusões e Perspectivas Futuras:** contém as conclusões e contribuições do trabalho, além de indicar a continuação da pesquisa através de trabalhos futuros.

## CAPÍTULO 2 - AVALIAÇÃO DE USABILIDADE DE MODELOS DE PROJETOS DE APLICAÇÕES *WEB*

### 2.1 Introdução

Segundo ROCHA e BARANAUSKAS (2003), uma avaliação de usabilidade tem três grandes objetivos: avaliar a usabilidade das funções do sistema (se está adequada aos requisitos da tarefa do usuário), avaliar o efeito da interface junto ao usuário (avaliar sua usabilidade) e identificar problemas específicos do sistema (aspectos do design os quais quando usados no contexto alvo, causam resultados inesperados ou confusão entre os usuários). Os métodos comumente adotados para avaliação de usabilidade podem ser divididos em duas grandes categorias: (1) Inspeções de Usabilidade, nas quais inspetores examinam aspectos da aplicação para detectar violações de princípios de usabilidade estabelecidos; e (2) Testes de Usabilidade, que são métodos de avaliação baseados na participação direta de usuários (PRATES e BARBOSA 2003). A seguir são apresentados os principais métodos para Avaliação de Usabilidade, divididos em dois grandes grupos: Inspeção e Teste de usabilidade.

### 2.2 Métodos de Avaliação de Usabilidade

A **Inspeção de Usabilidade** é uma avaliação onde não há a participação do usuário final e pode ser usada em qualquer fase do desenvolvimento do *software*. Segundo ROCHA e BARANAUSKAS (2003), esta objetiva encontrar problemas de usabilidade em um design de uma interface de usuário e com base nesses problemas fazer recomendações no sentido de eliminar os problemas e melhorar a usabilidade do design. Isso significa que inspeções de usabilidade são feitas em um estágio onde a interface está sendo gerada e necessita ser avaliada. Além disto, PRATES e BARBOSA (2003) ressaltam que este tipo de avaliação possibilita a formação e capacitação da equipe com relação a projetos de interface centrados no usuário. Entre os principais métodos de inspeção existentes estão:

- **Avaliação Heurística:** proposta por NIELSEN (1994a), envolve um pequeno conjunto de avaliadores examinando a interface e julgando suas características tomando como base princípios de usabilidade, denominados heurísticas (PRATES e BARBOSA 2003). Este método será apresentado na Subseção 2.2.1.
- **Percorso Cognitivo (*Walkthrough* Cognitivo):** é um método que tem como foco principal avaliar o design quanto à sua facilidade de aprendizagem, onde o

autor de um aspecto do design apresenta uma proposta para um grupo de pares e estes avaliam a solução usando critérios apropriados ao design específico (POLSON *et al.* 1992).

- **Método de Inspeção Semiótica:** É um método em que um especialista percorre a interface e identifica potenciais rupturas de comunicação que poderiam surgir na interação usuário-sistema (DE SOUZA *et al.* 2006).
- **Guidelines e Checklists:** segundo WINCKLER e PIMENTA (2002), são recomendações ergonômicas que podem ser usadas com o duplo propósito de auxiliar o processo de concepção ou guiar a avaliação. Durante o processo de concepção, designers devem consultar tais recomendações que auxiliam evitar problemas de usabilidade. E pode-se utilizar o conjunto de recomendações ergonômicas como suporte para a inspeção da interface; neste caso, a interface é inspecionada minuciosamente por um avaliador que verifica se todas as recomendações ergonômicas são respeitadas.

Dentro do grupo de inspeção de usabilidade, pode-se identificar um subgrupo de técnicas que utilizam modelos de como o usuário usaria o sistema proposto para obter medidas de usabilidade preditas por cálculo ou simulação (KIERAS 2009). Nesta dissertação, vamos chamar estas abordagens de modelos que predizem a usabilidade. Entre as mais relevantes abordagens estão:

- **Modelos de Redes de Tarefas (*Task network models*)** (LAUGHERY 1989);
- **Modelos de Arquitetura Cognitiva (*Cognitive architecture models*)** (ANDERSON 1983) e;
- **Modelos GOMS** (CARD *et al.* 1983).

Já os **Testes de Usabilidade** são métodos de avaliação centrados no usuário que incluem métodos experimentais, métodos observacionais e técnicas de questionamento. Para usar esses métodos é necessária a existência de uma implementação real do sistema em algum formato, que pode variar desde uma simulação da capacidade interativa do sistema, um protótipo básico implementado, um cenário ou a implementação completa do sistema (ROCHA e BARANAUSKAS 2003). Entre as técnicas utilizadas nos Testes de Usabilidade, pode-se destacar:

- **Técnicas baseadas em Observação:** são técnicas que têm por fundamento a observação do usuário interagindo com o sistema. Uma delas é *Thinking Aloud*, na qual se solicita ao usuário que verbalize tudo que pensa enquanto usa um sistema e

a expectativa é que seus pensamentos mostrem como o usuário interpreta cada item da interface (HORNBACK *et al.* 2007).

- **Técnicas baseadas em Perguntas:** são técnicas que coletam a opinião dos usuários sobre a interface. Entrevista é uma destas técnicas onde um entrevistador faz perguntas ao usuário e estas podem ser gravadas ou apenas registradas pelo entrevistador.
- **Avaliação através da Monitoração de Respostas Fisiológicas:** um exemplo deste tipo de avaliação é o uso de aparelhos de *Eye Tracking*, que determinam o movimento e a fixação dos olhos de uma pessoa (NAMAHN 2001) e isto reflete na quantidade de processamento cognitivo que um display da interface requer.
- **Avaliação Experimental:** avaliação de aspectos específicos através de Estudos Experimentais e Estudos de Observações. Um Estudo Experimental é um estudo direcionado por hipóteses e permite a análise quantitativa. Já um Estudo de Observação é um estudo onde é possível coletar dados sobre como a tecnologia é aplicada (MAFRA *et al.* 2006) e permite uma análise qualitativa.
- **Método de Avaliação de Comunicabilidade.** É um método que envolve a observação de usuário em um ambiente controlado por um especialista, onde este analisa a interação do usuário com o sistema, identificando então as rupturas vivenciadas pelos usuários (PRATES e BARBOSA 2007).

Embora o teste de usabilidade seja considerado o método mais eficaz para avaliar sistemas e protótipos do ponto de vista do usuário das aplicações, seu custo é alto, pois envolve o tempo dos usuários e muitas vezes o uso de laboratórios específicos de usabilidade (MATERA *et al.* 2002). Os métodos de inspeção foram propostos como uma alternativa com bom custo-benefício em comparação com os testes de usabilidade.

### 2.2.1 Avaliação Heurística

NIELSEN (1994b) propôs um método de inspeção de *software*, chamado Avaliação Heurística (AH) e este tem por objetivo encontrar problemas de usabilidade através de uma análise de conformidade do sistema frente a padrões de qualidade ou heurísticas definidas por especialistas. Heurísticas são princípios que podem guiar uma decisão de projeto ou serem utilizados para uma análise crítica de uma decisão tomada (NIELSEN 1994b).

MATERA *et al.* (2006) cita a AH como um dos principais métodos de inspeção de aplicações interativas. Além disso, estudos mostram que este é um método muito eficiente, com uma boa relação custo-benefício (NIELSEN 1994a).

A Tabela 1 apresenta as 10 Heurísticas de Nielsen desenvolvidas a partir do estudo de 249 problemas de usabilidade (NIELSEN 1994a):

**Tabela 1** - Heurísticas de NIELSEN (1994a).

Nº	Heurística	Explicação
1	Visibilidade do estado do sistema	O sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de um <i>feedback</i> apropriado dentro de um tempo razoável.
2	Concordância entre o sistema e o mundo real	O sistema deve utilizar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário. Seguir as convenções do mundo real e fazer a informação aparecer na ordem natural e lógica.
3	Controle e liberdade ao usuário	O sistema deve dar apoio a funções como <i>Undo</i> e <i>Redo</i> ou funções que permitam ao usuário utilizar “saídas de emergência” em caso de escolhas de funções erradas ou para sair de um estado não esperado.
4	Consistência e padrões	Devem ser seguidas convenções da plataforma de desenvolvimento e padrões de interface normalmente aceitos. Usuários não devem ter que adivinhar se palavras, situações ou ações diferentes significam a mesma coisa.
5	Prevenção de Erros	O sistema deve prevenir a ocorrência de erros na sua utilização. Melhor do que apresentar boas mensagens de erros, é ter um projeto cuidadoso que previne a ocorrência de um problema, em primeiro lugar.
6	Reconhecer ao invés de lembrar	Tornar objetos, ações e opções visíveis, para que o usuário não tenha que se lembrar de informações de uma parte do diálogo para outra. Instruções para uso do sistema devem estar visíveis, ou facilmente recuperáveis, quando necessário.
7	Flexibilidade e eficiência de uso	Aceleradores (abreviações, teclas de função...) podem tornar mais rápida a interação com o usuário. Permitir aos usuários customizar ações freqüentes.
8	Projeto minimalista e estético	Diálogos não devem conter informação irrelevante ou raramente necessária. Todas as unidades extras de informações em um diálogo competem com aquelas que são realmente relevantes, e diminuem sua visibilidade relativa
9	Reconhecimento, diagnóstico e recuperação de erros	Mensagens de erros devem ser expressas em linguagem simples (sem códigos), indicando precisamente o problema, e sugerindo construtivamente uma solução.
10	Ajuda e Documentação	As informações de ajuda e documentação devem ser fáceis de procurar, com foco na tarefa do usuário, listando passos concretos que devem ser seguidos e não serem grandes demais.

Segundo ROCHA e BARANAUSKAS (2003), a Avaliação Heurística deve ser vista como parte do processo de design interativo de uma interface. Esta envolve um pequeno conjunto de avaliadores examinando a interface e julgando suas características usando os princípios de usabilidade, descritos acima.

Na primeira fase, cada avaliador percorre a interface diversas vezes inspecionando os diferentes componentes do diálogo e ao encontrar problemas os descreve associando-os

as heurísticas de usabilidade que foram violadas, julgando-os também quanto à gravidade de acordo com a escala de severidade abaixo:

- 0 – Não representa um problema de usabilidade;
- 1 – Somente um problema cosmético. Não precisa ser consertado a não ser que haja tempo disponível;
- 2 – Problema leve. Baixa prioridade para consertá-lo;
- 3 – Problema grave. Alta prioridade para consertá-lo;
- 4 – Problema catastrófico. É obrigatório consertá-lo.

A segunda fase consiste de uma reunião onde é feita a consolidação das listas individuais de problemas dos avaliadores. Cada avaliador tem acesso aos relatórios individuais de todos os avaliadores, e podem julgar sobre os problemas encontrados pelos outros avaliadores. Ao final dessa fase relatório unificado é obtido com todos os problemas de usabilidade encontrados.

E na última fase, selecionam-se os problemas que devem ser corrigidos de acordo com a severidade, prazos e orçamentos do mesmo. Esta etapa deve ser realizada junto ao cliente ou ao gerente de projeto.

### 2.3 Avaliação de Usabilidade de Aplicações *Web*

Nos últimos anos, vários métodos têm sido propostos e/ou adaptados para aplicações *Web*, pois estas “são aplicações interativas, centradas no usuário e baseadas em hipermídia, onde a interface com o usuário desempenha um papel central” (OLSINA *et al.* 2006), onde certos aspectos característicos devem ser considerados, devido às especificidades destes tipos de aplicações. HITZ *et al.* (2006) citam duas características diferenciais que devem ser consideradas em relação à usabilidade de aplicações *Web*:

- **Inviabilidade (ou Dificuldade) de Treinamento dos Internautas:** alguns programas de treinamento são utilizados para compensar defeitos de usabilidade do produto, tentando adaptar os usuários às restrições da aplicação. Todavia, programas de treinamento não são viáveis para a maior parte das aplicações *Web*, disponíveis para uso por internautas. Por este motivo, há uma maior ênfase em tornar as aplicações *Web* auto-explicativas;
- **Anonimidade dos Internautas:** há dificuldades de fazer suposições sobre os usuários de uma aplicação *Web*, uma vez que não há um ambiente delimitado de uso, tornando difícil adequá-la às necessidades específicas dos usuários.

Segundo MATERA *et al.* (2006), devido à importância da usabilidade para as aplicações *Web*, a indústria de desenvolvimento de *software* está investindo em técnicas e ferramentas para projetos e avaliações que ajudem a melhorar esse atributo de qualidade em suas aplicações *Web*. FERNANDEZ *et al.* (2011b) apresentam um mapeamento sistemático sobre métodos de avaliação de usabilidade para Aplicações *Web*, no qual foram encontrados 81 artigos apresentando métodos especificamente criados para avaliações de aplicações *Web*. FERNANDEZ *et al.* (2011b) citam como exemplos representativos de métodos específicos para aplicações *Web*: o método *Cognitive Walkthrough for the Web* (CWW) (BLACKMON *et al.* 2002), a técnica *Web Design Perspective-based Inspection Technique* (WDP) (CONTE *et al.* 2009) e o método *Milano–Lugano Evaluation Method* (MiLE+) (TRIACCA *et al.* 2005). Porém, estes são métodos para avaliação da aplicação final ou de uma versão preliminar da interface. Na próxima seção serão apresentadas técnicas de inspeção propostas para avaliação de usabilidade a partir de modelos de análise e projeto.

## 2.4 Técnicas de Inspeção de Modelos

Como dito anteriormente, a avaliação de usabilidade geralmente ocorre no final do processo de desenvolvimento de *software*, e por este motivo alguns problemas de usabilidade deixam de ser encontrados e tratados logo no início do ciclo de desenvolvimento, onde são mais baratos de resolver. Identificar problemas de usabilidade cedo, no entanto, é difícil com a prática atual de desenvolvimento de *software*, porque o trabalho de usabilidade é geralmente separado das atividades do núcleo de desenvolvimento de *software* (HORNBÆK *et al.* 2007).

Recentemente algumas técnicas foram criadas para garantir a usabilidade nas etapas iniciais do processo de desenvolvimento de aplicações *Web* (FERNANDEZ *et al.* 2009b). Então, para incorporar a usabilidade ao processo de desenvolvimento de aplicações *Web* de maneira efetiva, é necessário caracterizar as técnicas já existentes, a fim de conhecer as etapas do processo de desenvolvimento que estas abordam assim como conhecer pontos que estas não cobrem.

Motivado por este contexto, foi realizado um levantamento das técnicas de inspeção de modelos de aplicações *Web* citados na literatura técnica. Com isso, foi feito um mapeamento sistemático da literatura (KITCHENHAM *et al.* 2010), apresentado a seguir e em dois Relatórios Técnicos, com a busca de artigos de periódicos (VALENTIM e CONTE 2011) e conferências relevantes da área (VALENTIM *et al.* 2012a). Um mapeamento sistemático é um tipo de revisão sistemática e segundo KITCHENHAM

(2004), é uma forma de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma questão de pesquisa particular.

#### **2.4.1 Mapeamento Sistemático sobre Técnicas de Inspeção de Modelos que garantam a Usabilidade de Aplicações *Web***

- **Objetivo**

O objetivo deste mapeamento foi examinar as propostas atuais de técnicas para inspeção de modelos de análise e projeto, para responder a seguinte questão de pesquisa: *“Quais técnicas de inspeção de modelos de análise e projeto têm sido propostas para garantir a usabilidade de aplicações Web nos estágios iniciais de desenvolvimento?”*. Além desta questão de pesquisa geral, foram definidas subquestões, para responder questionamentos específicos sobre a aplicabilidade de cada técnica. As subquestões são: (a) quais artefatos (modelos) são o foco de avaliação destas técnicas?; (b) quais técnicas requerem um apoio de uma ferramenta específica para sua aplicação?; e (c) a avaliação é feita através de uma técnica de inspeção de usabilidade em modelos ou através de uma técnica de avaliação baseada em modelos que predizem a usabilidade?.

- **Estratégia utilizada para pesquisa dos estudos primários**

A estratégia para pesquisa deve tornar explícito o escopo da pesquisa e os termos a serem utilizados na mesma, que são utilizados para compor as seqüências de palavras-chave para busca (*search strings*). Estes termos são definidos a partir da população, intervenção e resultados esperados definidos na questão de pesquisa.

- Escopo da pesquisa: pesquisa em bases de dados eletrônicas com acesso ao texto completo de artigos publicados em revistas e em anais de conferências selecionadas.
  - Fontes: Bibliotecas digitais com acesso ao texto completo de artigos através de seus engenhos de busca avançada e pesquisa manual em anais de conferências mais importantes da área no Brasil através de critérios que serão estabelecidos a seguir. A Tabela 2 mostra as fontes utilizadas. O Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC) foi escolhido por ser a conferência nacional mais importante da área. A pesquisa por artigos desta conferência foi feita nos anais dos anos de 2010 e 2011, devido a disponibilidade dos mesmos.

**Tabela 2** - Fontes utilizadas.

Nome da Fonte	Link
IEEE Xplore	<a href="http://ieeexplore.ieee.org/">http://ieeexplore.ieee.org/</a>
Scopus	<a href="http://www.scopus.com/home.url">http://www.scopus.com/home.url</a>
ACM Digital Library	<a href="http://portal.acm.org/dl.cfm">http://portal.acm.org/dl.cfm</a>
IHC 2010	<a href="https://www.ufmg.br/swib/?p=139">https://www.ufmg.br/swib/?p=139</a>
IHC 2011	<a href="http://www.cin.ufpe.br/~ihc_clihc2011/">http://www.cin.ufpe.br/~ihc_clihc2011/</a>

- Idiomas dos Artigos: os idiomas escolhidos foram o Inglês e o Português. O Inglês foi selecionado por ser adotado pela grande maioria das conferências e periódicos internacionais relacionados com tema de pesquisa e por ser o idioma utilizado pela maioria das editoras relacionadas com o tema, listadas no Portal de Periódicos da CAPES. O Português foi escolhido adicionalmente por ser um dos idiomas adotados nos artigos dos anais do Simpósio de Fatores Humanos em Sistemas Computacionais (IHC).
- Termos utilizados na pesquisa: os termos que foram utilizados neste mapeamento foram agrupados em quatro grupos que combinados entre si formam as *strings* de busca. Na Tabela 3 são mostrados os termos utilizados na pesquisa.

**Tabela 3** - Termos utilizados.

Grupo 1	Grupo 2	Grupo3	Grupo 4
web application	Development	usability inspection	model
web system	Project	usability evaluation	design
web software	Engineering	usability technique	
web-based software		usability method	
web-based system		usability inspection technique	
		usability inspection method	

Devido ao grande número de termos, foi decidido dividir a busca em várias seqüências de palavras-chave. Com isso, a complexidade da *string* de busca foi reduzida, o que garante melhor legibilidade e diminui a probabilidade de erro (humano e das máquinas de busca das editoras). As seqüências de palavras-chave utilizadas são listadas a seguir:

**Search 1a:** (usability inspection OR usability evaluation OR usability technique OR usability method) AND (model OR design) AND (web application development OR web application project OR web application);

**Search 1b:** (usability inspection OR usability evaluation OR usability technique OR usability method) AND (model OR design) AND (web system development OR web system project OR web system);

**Search 1c:** (usability inspection OR usability evaluation OR usability technique OR usability method) AND (model OR design) AND (web software development OR web software project OR web software).

- **Cr terios e procedimentos de sele o de estudos**

Seguindo os procedimentos descritos em (KITCHENHAM 2004), foram definidos crit rios de inclus o e exclus o para os artigos retornados pela *string* de busca.

Os crit rios definidos para inclus o de artigo foram:

- CI1. Podem ser selecionadas publica es que apresentam t cnicas ou m todos de avalia o de usabilidade aplicada a modelos do processo de desenvolvimento de aplica es *web*;
- CI2. Podem ser selecionadas publica es que descrevam um prot tipo de ambiente *web* desde que seja acompanhado de uma justificativa das caracter sticas deste ambiente em rela o   avalia o de usabilidade aplicada a modelos do processo de desenvolvimento deste ambiente;
- CI3. Podem ser selecionadas publica es que discutam aspectos relacionados a abordagens de avalia o de usabilidade aplicada a modelos do processo de desenvolvimento de aplica es *web*, incluindo apoio ferramental;
- CI4. Podem ser selecionadas publica es que apresentam uma abordagem usada em uma organiza o desenvolvedora de aplica es *web* para avaliar a usabilidade dos modelos do processo de desenvolvimento de suas aplica es *web*;
- CI5. Podem ser selecionadas publica es onde s o descritas ou apresentadas t cnicas ou m todos de avalia o de usabilidade aplicada a modelos do processo de desenvolvimento de aplica es *web* em tutoriais, cursos, workshops e similares.

Os crit rios definidos para exclus o de artigo foram:

- CE1. N o ser o selecionadas publica es em que as ferramentas citadas n o estejam relacionadas a t cnicas ou m todos de avalia o de usabilidade aplicada a modelos do processo de desenvolvimento de aplica es *Web*;
- CE2. N o ser o selecionadas publica es que n o tem disponibilidade de conte do para leitura e an lise dos dados (especialmente em casos, onde os estudos s o pagos ou n o disponibilizados pelas m quinas de buscas).

Quanto ao procedimento de sele o preliminar (1  Filtro), foi decidido que seriam selecionados artigos que apresentassem informa es no t tulo e no *abstract* sobre t cnicas de avalia o de usabilidade em modelos de processo de desenvolvimento de *software* utilizando

os critérios de inclusão e exclusão descritos anteriormente. Em caso de dúvida, o artigo seria incluído. Para cada estudo incluído ou excluído foi apresentada uma justificativa.

Em relação ao procedimento de seleção final (2º Filtro), como a estratégia de leitura de somente duas informações (título e *abstract*) não é suficiente para identificar se o estudo é realmente relevante, tornou-se necessário realizar a leitura completa dos artigos que restaram da seleção preliminar. Dessa forma, a presente etapa teve como objetivo fazer uma análise mais detalhada identificando e extraindo dados também de acordo com os critérios de inclusão e exclusão descritos anteriormente. Novamente, para cada estudo incluído ou excluído foi apresentada uma justificativa.

- **Estratégia para extração de dados**

De cada artigo aprovado pelo processo de seleção completo, foram extraídos os seguintes dados, de acordo com a Tabela 4:

**Tabela 4** - Tabela para extração de dados.

Nome	Referência (autor, ano)
<b>Especificação do tipo da técnica</b>	Descrição do tipo de técnica (inspeção ou teste)
<b>Baseia-se em algum método?</b>	Utiliza como base para formulação da técnica algum método existente
<b>Descrição da técnica</b>	Descrição da técnica utilizada na avaliação
<b>Contextualização</b>	Ambiente em que foi aplicada a técnica (academia, indústria, laboratório, etc.)
<b>Fase do processo para ser aplicado</b>	Fase do processo de desenvolvimento em que a avaliação é realizada – aplicação da técnica.
<b>Tem apoio ferramental</b>	Alguma ferramenta foi desenvolvida para dar suporte a técnica proposta
<b>Tipo de avaliação</b>	Resumo da avaliação de usabilidade efetuada. O que foi considerado defeito na aplicação e/ou problema de usabilidade, bem como o tipo de estudo realizado (estudo experimental, estudo de caso, prova de conceito, etc...)
<b>Limitações</b>	Limitações encontradas na aplicação da técnica e na análise dos estudos e execução da técnica.

#### 2.4.2 Artigos Selecionados Após a Condução o Mapeamento Sistemático

A Tabela 5 a seguir apresenta as quantidades de artigos retornados por cada máquina de busca das bibliotecas digitais selecionadas e quantidades de artigos dos anais do IHC de 2010 e 2011, o total de artigos que foram selecionados no 1º Filtro e o total de artigos selecionados após o 2º Filtro. Pode-se notar que a diferença de artigos retornados pela biblioteca ACM Digital Library e a quantidade de artigos que foram selecionados da mesma fonte é grande. Isto se deve parcialmente ao fato de ter existido um grande retorno com nomes de conferências e seus resumos estendidos no meio dos artigos.

**Tabela 5** - Totais de artigos retornados, selecionados no 1º filtro e selecionados após o 2º filtro.

Fonte	Total de Artigos Retornados	Total de Artigos Selecionados no 1º Filtro	Total de Artigos Selecionados no 2º Filtro
IEEEExplore	197	20	3
Scopus	46	13	1
ACM Digital Library	336	9	1
IHC 2010	66	5	2
IHC 2011	47	6	4
<b>Total</b>	692	53	11

Um ponto importante que precisa ser destacado aqui é que a biblioteca Scopus retornou um alto número de artigos sem acesso gratuito ao seu conteúdo através do portal de periódicos CAPES<sup>2</sup>, impossibilitando assim a leitura. Buscou-se entrar em contato com autores de artigos que se achavam importantes para a pesquisa e obteve-se resposta de apenas dois autores. Outro ponto importante é que houve artigos duplicados nas bibliotecas. Portanto, a ordem de execução do mapeamento sistemático nas bibliotecas seguiu-se de acordo com a Tabela 5. Conforme era apresentado um artigo duplicado da(s) biblioteca(s) anterior(es), este era descartado.

Além da mapeamento sistemático, usou-se uma técnica chamada *snowballing* para encontrar outros artigos. *Snowballing* é uma técnica que tem como objetivo identificar artigos citados a partir da lista de referências de outros artigos (JALALI e WÖHLIN 2012). Os artigos encontrados nesta pesquisa por *snowballing* foram identificados a partir das referências citadas nos artigos selecionados do mapeamento sistemático. Na Tabela 6 são listados artigos selecionados do mapeamento sistemático relevantes para esta pesquisa, juntamente com outros artigos encontrados através do *snowballing*. Para cada artigo é apresentada uma descrição sucinta das técnicas de inspeção de modelos que garantem a usabilidade de aplicações *Web*. Além disso, são indicados quais artigos foram identificados pelo mapeamento sistemático e quais foram por *snowballing*.

**Tabela 6** - Técnicas para inspeção de modelos que garantem a usabilidade.

Referência	Descrição da Técnica	Método
(ABRAHÃO e INSFRAN 2006) (FERNANDEZ 2009a) (FERNANDEZ <i>et al.</i> 2009b) (FERNANDEZ <i>et al.</i> 2010)	Apresenta o WUEP ( <i>Web Usability Evaluation Process</i> ), uma avaliação que tem como objetivo integrar as avaliações de usabilidade em MDWD ( <i>Model-Driven Web Development Processes</i> ), empregando um modelo de usabilidade <i>Web</i> como o artefato de entrada principal. Este modelo decompõe o conceito de usabilidade em sub-características e atributos mensuráveis, que são então associados a métricas, a fim de quantificá-los.	Mapeamento sistemático

<sup>2</sup> www.periodicos.capes.gov.br

Referência	Descrição da Técnica	Método
(FERNANDEZ <i>et al.</i> 2011a)	Essas métricas fornecem uma definição genérica e o que deve ser operacionalizado de forma a ser aplicável a artefatos de diferentes níveis de abstração em diferentes processos MDWD.	
(ATTERER 2005) (ATTERER e SCHMIDT 2005) (ATTERER <i>et al.</i> 2006) (ATTERER 2008)	Apresenta uma abordagem que melhora a usabilidade durante o desenvolvimento de <i>websites através de uma</i> ferramenta de usabilidade automatizada. Informações adicionais e o código HTML geram como saída um relatório de usabilidade que ajuda a melhorar <i>websites, a</i> partir de um modelo. Para isso, introduziu-se um novo metamodelo que permitiu anexar as informações de contexto para um modelo de engenharia <i>web</i> existente.	Mapeamento sistemático
(COSTA NETO <i>et al.</i> 2011)	Propõe uma avaliação de usabilidade por meio de inspeção de modelos de interação (descritos na linguagem ALaDIM – “ <i>Abstract Language for Description of Interactive Message</i> ”), realizado em um estilo checklist de um conjunto de diretrizes, constituídas com base nas heurísticas de Nielsen.	Mapeamento sistemático
(FERREIRA <i>et al.</i> 2010)	Apresenta uma abordagem onde práticas de desenvolvimento centradas no usuário são incorporadas em um processo de desenvolvimento de maneira que problemas associados à usabilidade sejam evitados. Na fase de análise e projeto aborda definição de padrões, definição de heurísticas de projeto de interface, dentre outros.	Mapeamento sistemático
(HORNBAEK <i>et al.</i> 2007)	Apresenta o UCE ( <i>Use Case Evaluation</i> ), uma avaliação de usabilidade baseado nas heurísticas de Nielsen que foi adaptado para avaliação de usabilidade baseada em casos de uso com o objetivo de facilitar a identificação dos problemas de usabilidade no ponto no processo de desenvolvimento, onde os primeiros casos de uso essenciais são descritos.	<i>Snowballing</i>
(LIANG e DENG 2009)	Apresenta um modelo de simulação de tarefa cooperativa (CTSM - <i>Cooperative Task Simulating Model</i> ) baseado em GOMS (Goals, Operators, Methods, and Selection rules) estendido, a partir do qual é construído um gráfico de roteamento de interação. O Modelo de Avaliação de Usabilidade se inicia no cenário, onde cada cenário é mapeado para cada uso da tarefa de alto nível, depois se extrai o modelo do contexto de uso semi-construído a partir deste. A inspeção de usabilidade precoce é iterativamente trabalhada em um protótipo não funcional, chamado gráfico de roteamento de interação, para descoberta de problemas de usabilidade.	Mapeamento sistemático
(LUNA <i>et al.</i> 2010)	A abordagem se baseia em requisitos de usabilidade funcional, chamados de <i>Functional Usability Features</i> (FUF) (JURISTO <i>et al.</i> 2007a, JURISTO <i>et al.</i> 2007b), que são requisitos de usabilidade relacionados à funcionalidade e, portanto, relacionados com a arquitetura do sistema. Esta abordagem propõe começar construindo teste de interação e navegação derivados de <i>mockups</i> de apresentação e diagramas de interação do usuário (UIDs). Estes testes são depois executados na aplicação gerada automaticamente pela	<i>Snowballing</i>

Referência	Descrição da Técnica	Método
	ferramenta de desenvolvimento dirigida por modelos utilizada pelo método em questão.	
(MOLINA e TOVAL 2009)	Apresenta uma avaliação da qualidade com foco em usabilidade em tempo de projeto para melhorar os modelos utilizados para o desenvolvimento de Sistemas <i>Web</i> de Informação - WIS ( <i>Web Information Systems</i> ). O modelo QMW ( <i>Quality Metrics Web</i> ) possui um conjunto de métricas para avaliar a qualidade dos artefatos usados no desenvolvimento WIS. Após a identificação dessas métricas, foi feita a integralização destas dentro da ferramenta que apóia a abordagem.	<i>Snowballing</i>
(PANACH <i>et al.</i> 2007)	Faz uso de dois padrões de usabilidade (Padrão Passo a Passo e Padrão Entrada de Texto Estruturado) com o objetivo de resolver problemas de usabilidade do modelo de apresentação do OOWS ( <i>Object Oriented Web Solutions</i> ) que não modelam estas funcionalidades. A solução proposta é a utilização de elementos estereotipados UML para estes padrões. Os novos elementos UML são introduzidos no Modelo de Apresentação do modelo OOWS.	Mapeamento sistemático
(SILVA e SILVEIRA 2008) (SILVA e SILVEIRA 2010)	Apresenta um método informal de avaliação em IHC, mais especificamente listas de verificação de diretrizes ( <i>checklists</i> ), para ser utilizado nas etapas iniciais do processo de desenvolvimento quando se tem Diagramas de Casos de Uso e Diagramas de Atividades do sistema a ser desenvolvido. A ideia central da proposta é que o próprio projetista possa testar suas ideias, utilizando essas diretrizes para avaliar um sistema interativo ainda na etapa de modelagem e sem a necessidade de um especialista da área.	Mapeamento sistemático
(SOMMARIVA <i>et al.</i> 2011)	Faz uso de um jogo chamado Usabilitygame que é caracterizado como um jogo educacional do estilo emulador, <i>single player</i> , desenvolvido para <i>Web</i> . O jogo tem como cenário uma empresa de desenvolvimento de <i>software</i> fictícia que, após enfrentar problemas devido à falta de usabilidade de seus produtos está contratando Engenheiros de Usabilidade. Assim o jogador (aluno) ao ser contratado assume o papel de Engenheiro de Usabilidade e deve buscar um bom desempenho em cada uma das fases do jogo, que possuem relação com as fases do ciclo de vida proposto por MAYHEW (1999).	Mapeamento sistemático
(TAO 2005)	Propõe uma abordagem que faz uso do diagrama de estado para a definição de um modelo de comportamento, onde neste, um estado composto representa uma tela e um sub-estado um modo de interação. Segundo os autores, as heurísticas de Nielsen podem ser aplicadas ao modelo de comportamento, mesmo que a parte visível ainda não esteja disponível.	Mapeamento sistemático
(THIEL <i>et al.</i> 2007)	Apresenta uma abordagem que permite integrar e desenvolver métricas de usabilidade durante o processo de desenvolvimento e estimar a usabilidade de sistemas interativos. Inicia a partir de um nível abstrato onde as métricas serão derivadas de forma adequada para ser	<i>Snowballing</i>

Referência	Descrição da Técnica	Método
	aplicadas sobre os modelos de transformações do Modelo EMODE e tenta prever a usabilidade, investigando modelos antes de eles serem transformados em código.	
(VASCONCELOS e BALDOCHI JÚNIOR 2011)	Propõe uma ferramenta, chamada USABILICS, que captura as interações que considera os atributos CSS ( <i>Cascading StyleSheet</i> ) dos elementos das páginas, permitindo tratá-los genericamente, o que facilita a definição de diversos caminhos alternativos para uma mesma tarefa. Para analisar os eventos coletados, propõe-se um algoritmo que avalia a similaridade de sequências de eventos, resultando em uma métrica da eficiência e da eficácia da usabilidade em interfaces <i>Web</i> .	Mapeamento sistemático
(VILLEGAS <i>et al.</i> 2011)	Propõe a definição de um processo para o desenvolvimento de sistemas usáveis. Este envolve atividades definidas de um modelo de processo de desenvolvimento de <i>software</i> que inclui um protótipo e uma avaliação de usabilidade em todas suas fases, chamado MPIu + a ( <i>Modelo de Proceso de la Ingeniería de la Usabilidad y Accesibilidad</i> ).	Mapeamento sistemático

### 2.4.3 Categorização das Técnicas Encontradas e Análise

O propósito para o mapeamento realizado foi a criação de uma base de conhecimento que apóie a adoção das técnicas de inspeção de modelos de análise e projeto pela indústria de *software*. Para isto é necessário caracterizar quais técnicas podem ser utilizadas em quais processos de desenvolvimento.

Para que uma organização de desenvolvimento de *software* passe a adotar uma determinada técnica de inspeção de modelos, é necessário que esta organização, durante o seu processo, elabore os modelos que são avaliados pela técnica escolhida. Além disso, é preciso responder a questões que influenciam na possibilidade de adoção das técnicas propostas pela indústria de *software*. Por exemplo, técnicas que requerem o uso de determinadas ferramentas de apoio também restringem a sua adoção por organizações que possam adquirir e/ou incorporar estas ferramentas aos seus processos de desenvolvimento.

Para isso, foram propostas questões específicas apresentadas na Tabela 7.

**Tabela 7** - Questões Específicas para Caracterização das Técnicas de Inspeção de Modelos

Questão	Critério para Análise
Q1 - Quais artefatos (modelos) são o foco de avaliação destas técnicas?	Identificar quais modelos de análise e projeto estão sendo utilizados para avaliação de usabilidade
Q2 - Quais técnicas fazem uso de uma ferramenta específica para sua aplicação?	Para esta questão, busca-se analisar se a técnica requer (obrigatoriamente) ou sugere (opcionalmente) uma ferramenta de apoio à inspeção.
Q3 - A avaliação é feita	Identificar se a avaliação de usabilidade é feita através de uma

Questão	Critério para Análise
através de uma técnica de inspeção de usabilidade em modelos ou através de uma técnica de avaliação baseada em modelos que predizem a usabilidade?	técnica de inspeção de usabilidade em modelos ou através de uma técnica de avaliação baseada em modelos que predizem a usabilidade.

A seguir é apresentado o tipo de informação coletada para cada questão específica:

- **Foco na Avaliação** (Qual artefato?): cita em qual (quais) artefato(s) (modelo(s)) a técnica faz a avaliação de usabilidade.
- **Apoio Computacional** (Requer apoio específico/Sugere Apoio específico/Não): aponta se o processo exige, sugere ou não tem apoio computacional de ferramenta específica.
- **Avaliação de Usabilidade** (Inspeção de usabilidade em modelos/Avaliação baseada em modelos que predizem a usabilidade): indica se a avaliação é feita através de uma técnica que inspeciona a usabilidade em modelos ou se a avaliação é baseada em modelos que predizem a usabilidade.

Na Tabela 8 pode ser encontrada a classificação das técnicas identificadas de acordo com os critérios propostos:

**Tabela 8** - Classificação das Técnicas em relação às Questões Específicas

Referências para Artigos das Técnicas	Foco na Avaliação	Apoio Computacional	Avaliação de Usabilidade
(ABRAHÃO e INSFRAN 2006) (FERNANDEZ 2009a) (FERNANDEZ <i>et al.</i> 2009b) (FERNANDEZ <i>et al.</i> 2010) (FERNANDEZ <i>et al.</i> 2011a)	Modelos Independentes de Plataforma (PIM) e Modelos Específico de Plataforma (PSM)	Sugere Apoio específico (1)	Avaliação baseada em modelos que predizem a usabilidade
(ATTERER 2005) (ATTERER e SCHMIDT 2005) (ATTERER <i>et al.</i> 2006) (ATTERER 2008)	Diagrama de Atividades (estudos iniciais) e Metamodelo de usabilidade	Requer Apoio específico (2)	Inspeção de usabilidade em modelos
(COSTA NETO <i>et al.</i> 2011)	Modelo ALADIM	Não	Avaliação baseada em modelos que predizem a usabilidade
(FERREIRA <i>et al.</i> 2010)	Contexto de uso, protótipo, mapas de navegação	Não	Avaliação baseada em modelos que

Referências para Artigos das Técnicas	Foco na Avaliação	Apoio Computacional	Avaliação de Usabilidade
			predizem a usabilidade
(HORNBAEK <i>et al.</i> 2007)	Descrição dos Casos de Uso	Não	Inspeção de usabilidade em modelos
(LIANG e DENG 2009)	Cenários e Protótipo não-funcional chamado de Gráfico de Roteamento de Interação do Modelo CTSM	Não	Avaliação baseada em modelos que predizem a usabilidade
(LUNA <i>et al.</i> 2010)	<i>Mockups</i> de apresentação e diagramas de interação do usuário	Requer apoio específico (3)	Inspeção de usabilidade em modelos
(MOLINA e TOVAL 2009)	Modelos de Navegação do Modelo QMW	Requer apoio específico (3)	Avaliação baseada em modelos que predizem a usabilidade
(PANACH <i>et al.</i> 2007)	Elementos estereotipados da UML no Modelo de Apresentação do Modelo OOWS	Sugere Apoio específico (4)	Avaliação baseada em modelos que predizem a usabilidade
(SILVA e SILVEIRA 2008) (SILVA e SILVEIRA 2010)	Diagrama de Casos de Uso e Diagrama de Atividades	Não	Inspeção de usabilidade em modelos
(SOMMARIVA <i>et al.</i> 2011)	Modelo Conceitual e Protótipo	Requer Apoio específico (5)	Avaliação baseada em modelos que predizem a usabilidade
(TAO 2005)	Diagrama de Estado	Requer apoio específico (6)	Inspeção de usabilidade em modelos
(THIEL <i>et al.</i> 2007)	Modelo de transformação do Modelo EMODE	Requer apoio específico (7)	Avaliação baseada em modelos que predizem a usabilidade
(VASCONCELOS e BALDOCHI JÚNIOR 2011)	Modelo de interface	Requer Apoio específico (8)	Avaliação baseada em modelos que predizem a usabilidade
(VILLEGAS <i>et al.</i> 2011)	Modelo de Processo da Engenharia de Usabilidade e Acessibilidade	Requer Apoio específico (9)	Avaliação baseada em modelos que predizem a usabilidade

(1) A ferramenta VisualWade (Fernandez et al. 2009b), oferece suporte para edição e compilação dos modelos propostos pelo método OO-H (*Object-Oriented Hypermedia*).

Referências para Artigos das Técnicas	Foco na Avaliação	Apoio Computacional	Avaliação de Usabilidade
<p>(2) A ferramenta WUSAB (ATTERER <i>et al.</i> 2006), onde o desenvolvedor pode digitar a URL de uma página <i>web</i> para fazer a validação, gerando após isso um relatório de usabilidade que é apresentado ao desenvolvedor quando os problemas são identificados.</p> <p>(3) A abordagem é baseada no uso do framework i* (YU 1997).</p> <p>(4) A ferramenta OlivaNova (PANACH <i>et al.</i> 2007) gera o código referente a interface do usuário orientada a <i>Web</i>, pelo método OOWS.</p> <p>(5) Usabilitygame é caracterizado como um jogo educacional do estilo emulador, desenvolvido para <i>Web</i> (SOMMARIVA <i>et al.</i> 2011).</p> <p>(6) A abordagem requer uma ferramenta que ainda não foi desenvolvida, pois o trabalho está em andamento. Esta ferramenta será para capturar traços de eventos de interface do usuário e verificar se o que aconteceu com a interface de usuário é o especificado nos diagramas de estado (TAO 2005).</p> <p>(7) O framework (THIEL <i>et al.</i> 2007) é baseado em Java e usa os modelos eMODE, mas pode ser aplicada a qualquer metamodelo, que inclui um modelo de tarefa e um modelo de interface do usuário.</p> <p>(8) USABILICS é uma ferramenta para captura das interações que considera os atributos CSS (<i>Cascading StyleSheet</i>) dos elementos das páginas (VASCONCELOS e BALDOCHI JÚNIOR 2011).</p> <p>(9) Ferramenta que integra CIAF (<i>Collaborative Interactive Application Framework</i>) e MPIu+a (<i>Modelo de Proceso de la Ingeniería de la Usabilidad y Accesibilidad</i>) (VILLEGAS <i>et al.</i> 2011).</p>			

Após a categorização dos processos, foi possível realizar a seguinte análise em relação a cada critério:

Sobre o Foco na Avaliação, os quinze artigos focaram na avaliação de 21 diferentes modelos: Modelos Independentes de Plataforma; Modelos Específico de Plataforma; Descrição dos Casos de Uso; Diagrama de Casos de Uso; Diagrama de Atividades; Metamodelo de usabilidade; Contexto de uso; Protótipo; Mapas de navegação; Cenários de Interação; Gráfico de Roteamento de Interação; Modelo de Apresentação contendo elementos estereotipados da UML; *Mockups* de apresentação; Diagramas de interação do usuário; Diagrama de Estado; Modelos de Navegação e Modelo de Transformação de requisitos informais em especificações funcionais formais; Modelo ALaDIM; Modelo Conceitual; Modelo de Interface; Modelo de Processo da Engenharia de Usabilidade e Acessibilidade. É importante notar que técnicas que inspecionam modelos específicos, com notação própria ou propostos a partir de estereótipos, impõem à organização a inclusão desses modelos em seus processos.

Sobre o Apoio Computacional para a Inspeção: cerca de oito dos artigos apontaram que as técnicas requerem apoio computacional específico para sua execução restringindo assim significativamente a adoção do processo sugerido apenas por organizações com

acesso ao *framework*/ferramenta computacional em questão, dois dos artigos sugeriram apoio computacional e cinco não exigiram apoio computacional.

Em relação à Avaliação de Usabilidade: cinco das quinze técnicas realizam Inspeção de Usabilidade em Modelos e dez fazem Avaliação baseada em modelos que predizem a usabilidade, onde alguns dos modelos que estas avaliações se baseiam são: Modelo MDWD, Modelo CTSM, Modelo QMW, Modelo OOWS, Modelo EMODE, Modelo ALaDIM e Modelo de Processo da Engenharia de Usabilidade e Acessibilidade.

Espera-se que os resultados desta análise possam ser utilizados pelas organizações de desenvolvimento *Web* que estejam interessadas em garantir a usabilidade durante o desenvolvimento de suas aplicações. As informações sobre as técnicas existentes para inspecionar a usabilidade de modelos podem ser utilizadas para auxiliar na decisão de qual técnica se adéqua melhor ao seu projeto.

## 2.5 Considerações Finais

Este capítulo teve como objetivo apresentar os tipos de avaliações de usabilidade existentes assim como as específicas para as etapas iniciais do processo de desenvolvimento de aplicações *Web*.

Este capítulo também descreveu os resultados de um estudo secundário (mapeamento sistemático) realizado com o propósito de caracterizar e analisar as técnicas de inspeção de modelos de análise e projeto que garantam a usabilidade de aplicações *Web*. Esta caracterização mostrou, para cada técnica de inspeção encontrada, em qual artefato é feita a avaliação de usabilidade, se a inspeção de usabilidade requer apoio computacional de ferramenta específica e a qual tipo de avaliação de usabilidade se refere a abordagem.

O próximo capítulo apresenta os métodos para inspeção de usabilidade de modelos de análise e projeto selecionados como base para o conjunto proposto. Além disso, o capítulo mostra a proposta inicial de um conjunto de técnicas para a inspeção de modelos de análise e projeto para avaliação de usabilidade. Este conjunto é formado pela customização de técnicas já existentes e proposição de novas técnicas. O propósito deste conjunto de técnicas é que seja facilmente adotável pela indústria de *software*. Para tal, estas técnicas devem focar na avaliação de modelos em linguagens de notação padrão para projeto de sistemas e não requerer obrigatoriamente apoio de ferramentas específicas para a realização da avaliação de usabilidade.

## CAPÍTULO 3 - PROPOSTA INICIAL DO CONJUNTO DE TÉCNICAS MIT

### 3.1 Introdução

O aumento crescente da complexidade das aplicações *Web* e de suas características motivou a necessidade do uso de boas práticas no desenvolvimento de aplicações *Web* para que estas possam ser entregues dentro do prazo e orçamento planejados e com alto grau de qualidade e facilidade de uso e manutenção (PASTOR 2004). Isto tem se tornado uma questão crucial para a pesquisa em duas grandes áreas científicas: Engenharia de *Software* (ES) e Interação Humano-Computador (IHC). A Engenharia de *Software* tem se preocupado com as questões de definição de métodos de concepção (projeto), com o desenvolvimento das aplicações (e suas interfaces) e sua avaliação, já a área de IHC considera todos esses aspectos sobre a ótica humana, com a preocupação com a aceitabilidade social e prática das aplicações (GRISLIN e KOLSKI 1996). Sendo assim, a pesquisa de técnicas de inspeção deve focar em critérios de qualidade relevantes para o desenvolvimento de aplicações *Web* e considerar as características dos modelos destas aplicações segundo uma perspectiva não apenas de engenharia, mas de preocupação com a interação do usuário e o sistema.

Este trabalho foca nos modelos elaborados durante as atividades de análise e projeto das aplicações *Web*. Os modelos especificados na análise fornecem informações sobre “o quê” tem que ser feito no sistema. Já os modelos elaborados durante o projeto mostram “como” isso será feito. A vantagem de se encontrar problemas de usabilidade já nesses modelos está relacionada ao custo-benefício, já que resolvê-los no início do processo de desenvolvimento é mais barato e rápido do que resolvê-los quando o sistema já estiver pronto e em uso.

O objetivo principal deste trabalho é propor um conjunto de técnicas que garantam a usabilidade de aplicações *Web* através da avaliação dos modelos de análise e projeto. Como passo inicial para esta proposta, foi feita uma análise das técnicas já existentes para inspeção de modelos para garantia da qualidade (apresentadas previamente na Subseção 2.4.3).

Este capítulo apresenta as técnicas selecionadas que serviram como base para compor o conjunto de técnicas proposto neste trabalho, assim como as customizações e extensões sugeridas para melhor adoção pela indústria de desenvolvimento de *software*. A

Seção 3.2 aborda sobre a análise da viabilidade de uso das técnicas encontradas no Mapeamento Sistemático apresentado no Capítulo 2. A Seção 3.3 apresenta a descrição das técnicas selecionadas na análise de viabilidade de uso. A Seção 3.4 descreve a proposta inicial da técnica MIT (*Model Inspection Technique for Usability Evaluation*).

### 3.2 Análise da Viabilidade de Uso das Técnicas

Uma análise de viabilidade de uso das técnicas foi realizada para verificar quais técnicas poderiam ser útil para esta proposta. Todas estas técnicas já foram apresentadas no Capítulo 2 (Subseção 2.4.3).

A Tabela 9 a seguir apresenta uma análise de viabilidade de uso destas técnicas, ou seja, o motivo de ter contribuído ou não com esta proposta.

**Tabela 9** - Análise da Viabilidade de Uso das Técnicas

Técnicas	Análise da Viabilidade de Uso das Técnicas	Técnicas Selecionadas
(ABRAHÃO e INSFRAN 2006) (FERNANDEZ 2009a) (FERNANDEZ <i>et al.</i> 2009b) (FERNANDEZ <i>et al.</i> 2010) (FERNANDEZ <i>et al.</i> 2011a)	Esta técnica foca em um modelo que decompõe o conceito de usabilidade em subcaracterísticas e atributos mensuráveis, que são associados com métricas para quantificá-los. No entanto essas métricas não são diretamente associadas a defeitos de usabilidade em modelos de projeto. Por esta razão, seu uso foi considerado não viável no contexto desta proposta.	
(ATTERER 2005) (ATTERER e SCHMIDT 2005) (ATTERER <i>et al.</i> 2006) (ATTERER 2008)	A abordagem proposta, que tem o objetivo de usar modelos abstratos de abordagens de engenharia <i>web</i> durante a validação automática de usabilidade, antes de ser utilizada passou por um estudo para verificar problemas de usabilidade em Diagramas de Atividades que visa mostrar o fluxo de atividades em um único processo. Porém neste estudo o Diagrama de Atividade serviu apenas como base para a criação manual de adequados <i>links</i> e conteúdo da página. Com este exemplo, teve-se a ideia de avaliar a usabilidade através do Diagrama de Atividades, contribuindo assim para esta proposta.	
(COSTA NETO <i>et al.</i> 2011)	Esta abordagem apresenta uma avaliação de usabilidade por meio de inspeção de modelos de interação, que possui um conjunto de diretrizes baseadas nas heurísticas de Nielsen. Por este motivo, considerou-se que o uso desta técnica não seria viável no contexto desta pesquisa, pois o foco desta não são modelos de interação.	
(FERREIRA <i>et al.</i> 2010)	A abordagem descreve práticas de	

Técnicas	Análise da Viabilidade de Uso das Técnicas	Técnicas Selecionadas
	desenvolvimento centradas no usuário que agrega usabilidade ao processo de desenvolvimento para que problemas sejam evitados. No entanto, as práticas citadas, tanto na fase de análise quanto na fase de projeto, não são detalhadas no decorrer do artigo, impossibilitando uma possível análise e aproveitamento.	
(HORNBAEK <i>et al.</i> 2007)	Esta técnica aborda a avaliação de usabilidade em casos de uso, que frequentemente estão disponíveis no início do desenvolvimento de um sistema interativo, e são relevantes tanto para o desenvolvimento de <i>software</i> quanto para o design de interface de usuário. Por esta razão, o uso desta técnica foi considerado viável no contexto desta proposta.	
(LIANG e DENG 2009)	Esta técnica propõe a utilização de uma extensão do modelo GOMS, com um gráfico de roteamento. No entanto, não há evidências de facilidade de uso e utilidade deste gráfico de roteamento para justificar sua adoção por organizações de desenvolvimento de <i>software</i> . Desta forma, considerou-se que o uso desta técnica não seria viável no contexto desta pesquisa.	
(LUNA <i>et al.</i> 2010)	Esta abordagem faz teste de interação a partir de <i>mockups</i> de apresentação, que são artefatos usados para aspectos da Interface do Usuário que funcionam como esboços das aplicações. Estes são desenvolvidos nas fases iniciais para refletir os desejos dos clientes, onde podem ajudar a encontrar problemas de usabilidade mais cedo. Por este motivo, a ideia de avaliar <i>mockups</i> foi considerada viável no contexto desta proposta.	
(MOLINA e TOVAL 2009)	Esta técnica utiliza o modelo QMW ( <i>Quality Metrics Web</i> ) que possui um conjunto de métricas para avaliar a qualidade dos artefatos usados no desenvolvimento WIS ( <i>Web Information Systems</i> ) que são aplicadas sobre os modelos de navegação. Desta forma, considerou-se que o uso desta técnica não seria viável no contexto desta pesquisa, pois o foco desta não são modelos de navegação.	
(PANACH <i>et al.</i> 2007)	Esta técnica faz o uso de padrões, que é uma solução abstrata para um problema de usabilidade, sem levar em conta restrições de plataforma. Além disso, os padrões de projeto estão voltados para a implementação. Com isso, considerou-se que o uso desta técnica não seria viável no contexto desta pesquisa.	

Técnicas	Análise da Viabilidade de Uso das Técnicas	Técnicas Selecionadas
(SILVA e SILVEIRA 2008) (SILVA e SILVEIRA 2010)	Estas técnicas abordam avaliação de usabilidade em diagramas de casos de uso e diagramas de atividades, onde os projetistas obtêm um <i>feedback</i> informal dos usuários ou avaliadores para confirmar suas ideias. Os autores desse artigo realizaram estudos com estas técnicas e constataram que o diagrama de casos de uso tal como foi utilizado no experimento não permite avaliar a usabilidade de sistemas ainda na etapa de projeto, visto que neste diagrama não foi possível antecipar a identificação de nenhum problema de usabilidade, tampouco mapear algum dos problemas identificados na avaliação sobre o sistema. Já o diagrama de atividades possibilitou obter o fluxo de atividades que são relevantes para o desenvolvimento de <i>software</i> , assim como ajudou a encontrar problemas de usabilidade. Por esta razão, o uso da avaliação de usabilidade em diagramas de atividade foi considerado viável no contexto desta proposta.	
(SOMMARIVA <i>et al.</i> 2011)	Este jogo se baseia no ciclo de vida de engenharia de usabilidade proposto por MAYHEW (1999), que apresenta uma abordagem sobre engenharia de usabilidade. Esta possui as seguintes etapas: Análise de Requisitos, Projeto, Testes, Implementação e Instalação. O artigo não detalha como é feita a detecção dos problemas de usabilidade. Além disso, não foi possível utilizar o jogo, pois é necessário um cadastramento e não se conseguiu concluí-lo. Por esta razão, seu uso foi considerado não viável no contexto desta proposta.	
(TAO 2005)	Esta técnica sugere avaliação de usabilidade através do diagrama de estados e para isso é necessário capturar traços de eventos de interface do usuário (isto é, o que aconteceu no sistema como um resultado do usuário interagir com o sistema através de sua interface de usuário) e ver se o que aconteceu com a interface de usuário é especificado nos diagramas de estado. Mas isso não pode ser feito sem apoio automático. Por esta razão, seu uso foi considerado não viável no contexto desta proposta.	
(THIEL <i>et al.</i> 2007)	Esta técnica permite integrar e desenvolver métricas de usabilidade durante o processo de desenvolvimento para serem aplicadas sobre os modelos de transformações. No entanto essas métricas não são diretamente associadas a defeitos de usabilidade em	

Técnicas	Análise da Viabilidade de Uso das Técnicas	Técnicas Selecionadas
	modelos de projeto. Por esta razão, seu uso foi considerado não viável no contexto desta proposta.	
(VASCONCELOS e BALDOCHI JÚNIOR 2011)	Esta abordagem de avaliação de usabilidade de interfaces <i>Web</i> é composta por um modelo de interface para definição e análise de tarefas que é aplicado em três etapas: (1) definição das tarefas, (2) coleta automática dos dados dos eventos das interações dos usuários, e (3) análise de tarefas. Para analisar os eventos coletados, utiliza-se avaliação remota e automática de usabilidade. Por este motivo, considerou-se que o uso desta abordagem não seria viável no contexto desta pesquisa.	
(VILLEGAS <i>et al.</i> 2011)	A abordagem MPIu+a ( <i>Modelo de Proceso de la Ingeniería de la Usabilidad y Accesibilidad</i> ) considera a usabilidade como um atributo de qualidade no desenvolvimento de sistemas interativos e integra modelos e tarefas necessárias para suportar a avaliação de usabilidade em todo o ciclo de vida de ES. No entanto, o objetivo do artigo não é detalhar esta abordagem, e sim mostrar a integração deste com o CIAF ( <i>Collaborative Interactive Application Framework</i> ). Além disso, buscou-se encontrar os modelos de avaliação de usabilidade do MPIu+a e estes não estavam disponível. Com isso, considerou-se que o uso desta técnica não seria viável no contexto desta pesquisa.	

O método *Use Case Evaluation* (HORNBAEK *et al.* 2007) tem como base a Avaliação Heurística (AH) que será apresentado na Seção 3.3. O UCE (*Use Case Evaluation*) foi selecionado pelo fato deste apresentar uma avaliação de usabilidade em casos de uso e por também ser baseado nas heurísticas de Nielsen. Os casos de uso têm sido sugeridos como um meio valioso para a integração de engenharia de usabilidade diretamente no processo de desenvolvimento de *software* (FERRÉ *et al.* 2001). Os casos de uso são frequentemente disponíveis no início do desenvolvimento de um sistema interativo, e são relevantes tanto para o desenvolvimento de *software* quanto para o design de interface de usuário.

LUNA *et al.* (2010) apresenta uma abordagem que faz teste de interação a partir de *mockups* de apresentação. Estes *mockups* são artefatos usados para aspectos da Interface do Usuário que funcionam como esboços das aplicações. Destinam-se a ser desenvolvidos rapidamente para refletir os desejos dos clientes em termos de apresentação de forma mais substancial do que os requisitos expressos em linguagem escrita (LUNA *et al.* 2010). Por

este motivo, esta abordagem também foi selecionada como referencial para o conjunto de técnicas proposto, contribuindo apenas com a ideia de avaliação de usabilidade em *mockups*.

E por fim, ATTERER (2005) elaborou um estudo de caso para verificar problemas de usabilidade nos modelos de apresentação e navegação. Neste estudo criaram-se Diagramas de Atividades que tiveram como objetivo mostrar o fluxo de atividades em um único processo. Este diagrama da UML mostra como as atividades dependem uma da outra. E por esse motivo torna-se modelo onde é importante procurar por problemas de usabilidade. A abordagem de ATTERER (2005) buscou usar modelos abstratos de abordagens de engenharia *web* durante a validação automática de usabilidade e por isso o Diagrama de Atividade serviu apenas como base para a criação manual de adequados *links* e conteúdo da página. Porém, o estudo feito por ATTERER (2005) inspirou a proposta de uma técnica de avaliação de usabilidade em Diagramas de Atividade, assim como a abordagem de SILVA e SILVEIRA (2010) que se utilizou de um *checklist* para avaliação de usabilidade a partir dos Diagramas de Atividades.

### 3.3 Descrição das Técnicas Selecionadas

A seguir será feita uma descrição sucinta das quatro técnicas que contribuíram como base para esta proposta.

#### 3.3.1 Avaliação de Usabilidade em Casos de Uso

HORNBÆK *et al.* (2007) propuseram um método de avaliação chamado *Use Case Evaluation* (Avaliação de Casos de Uso) adaptado para avaliação de usabilidade baseada em casos de uso. Este método tem o objetivo de facilitar a identificação dos problemas de usabilidade no início do processo de desenvolvimento, onde os primeiros casos de uso essenciais são descritos.

O método de Avaliação de Casos de Uso (UCE) é composto por três atividades: (1) Inspeção de casos de uso, (2) Avaliação dos casos de uso, e (3) Documentação de avaliação.

Na primeira atividade o objetivo principal é identificar problemas de usabilidade que o avaliador está convencido que um usuário pode vir a experimentar. Esta atividade possui duas etapas:

- *Brainstorm*: o avaliador percorre os casos de uso, procurando por problemas de usabilidade, sem qualquer procedimento sistemático.

- **Inspeção Sistemática:** o avaliador tenta prever os problemas de usabilidade empregando as Heurísticas de Nielsen adaptadas (Tabela 10) buscando encontrar semelhanças em que um caso de uso ultrapasse uma heurística.

Nesta atividade as listas geradas são mescladas e formam uma lista com problemas de usabilidade.

**Tabela 10** - Heurísticas do Método UCE propostas por HORNBAEK et al. (2007).

<b>Nº</b>	<b>Heurística</b>	<b>Explicação</b>	<b>Baseado em</b>
1	Visibilidade do estado do sistema	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
2	Concordância entre o sistema e o mundo real	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
3	Controle e liberdade ao usuário	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
4	Consistência e padrões	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
5	Prevenção de Erros	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
6	Reconhecer ao invés de lembrar	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
7	Flexibilidade e eficiência de uso	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
8	Reconhecimento, diagnóstico e recuperação de erros	Mesma descrição da Avaliação Heurística	Heurísticas de Nielsen
9	Evitar operações mentais pesadas e menor carga de trabalho	Não forçar o usuário a operações mentais pesadas e manter a carga de trabalho do usuário no mínimo.	Heurísticas de Nielsen e Princípios de Dimensões Cognitivas
10	Evite forçar o usuário a compromissos prematuros	Não forçar o usuário a executar uma determinada tarefa ou decisão até que seja necessário. Será que o usuário sabe por que algo deve ser feito?	Heurísticas de Nielsen e Princípios de Dimensões Cognitivas
11	Proporcionar funções que são úteis para o usuário	Considerar se a funcionalidade descrita é útil para ser usada pelos usuários e se funções/dados estão faltando.	Ideia de que as primeiras avaliações em casos de uso devem ajudar a estabelecer o uso das funcionalidades propostas.

A segunda atividade tem o objetivo de avaliar a qualidade dos casos de uso, onde o avaliador expressa o porquê de algo não poder ser adequadamente analisado. Isso produz como resultado uma avaliação de cada caso de uso, que enfatiza o quão útil é o caso de uso para a inspeção.

E, por fim, na terceira atividade os resultados são compilados em um produto de avaliação coerente. O conteúdo principal da documentação é a lista de problemas de usabilidade, que descreve os problemas que os avaliadores esperam que um usuário vá enfrentar ao usar o sistema, sendo que cada um desses problemas de usabilidade devem incluir um motivo claro do porquê é percebido como tal.

Pode-se observar que duas Heurísticas de Nielsen foram excluídas da proposta de HORNBæk *et al.* (2007): “Projeto minimalista e estético” e “Ajuda e Documentação”. Isto se deve ao fato de que, segundo os autores, na especificação dos casos de uso não se têm a preocupação com detalhes do design da interface do usuário.

### 3.3.2 Avaliação de Usabilidade em *Mockups*

LUNA *et al.* (2010) apresentam um abordagem de desenvolvimento que visa ser ágil, *model-based development approach* (MDSB) e apoia a especificação de requisitos e testes de usabilidade. A abordagem começa construindo testes de interação e navegação derivados de *mockups* de apresentação e *User Interaction Diagrams* (UIDs). Estes testes são depois executados na aplicação gerada pela ferramenta de desenvolvimento dirigida por modelo para verificar se são aceitos ou não.

O ciclo de desenvolvimento é dividido em *sprints*. No início do *sprint*, a equipe de desenvolvimento tem apenas um conjunto de especificações do que eles têm que fazer. Estas especificações foram definidas por meio de entrevistas com o usuário. Desenvolvedores começam a trabalhar, escolhendo uma delas de cada vez. As etapas deste ciclo são descritas a seguir:

- Etapa 1: Um pequeno ciclo começa com a captura de uma análise mais detalhada usando artefatos de requisitos informal. Uma variedade de artefatos podem ser utilizados dependendo do tipo de requisito que se está captando:
  - Para requisitos envolvendo interações, usa-se UIDs que servem como uma especificação parcial da navegação da aplicação. *Mockups* são usados para aspectos da *User Interface* (UI), e *Use Case* (UC) ou *User Stories* (US) para aspectos do negócio ou de domínio.

- Para os requisitos de usabilidade, usa-se um conjunto de propriedades de usabilidade derivadas de orientações de requisitos de usabilidade definidas na literatura. Estas propriedades de usabilidade são representadas por: UIDs para aspectos de navegação e *mockups* para aspectos UI.
- Etapa 2: Os artefatos que foram usados para capturar os requisitos são descritos em linguagem informal, portanto, os desenvolvedores transformam esses requisitos em testes. Antes que um requisito seja implementado, os testes correspondentes devem ser executados para verificar se a aplicação cumpre ou não com os requisitos. A falta de testes mostra que requisitos ainda não são suportados pelo sistema em desenvolvimento. Se nesta fase o aplicativo passar por todos os testes, então, ou eles não expressam o novo requisito de forma adequada, ou o novo requisito não é novo na realidade, porque a aplicação já o implementa. No primeiro caso, deve-se detalhar mais os testes e voltar a etapa 2 e no segundo, devemos descartar o requisito e retornar a etapa 1.
- Etapa 3: Uma vez que o requisito foi especificado em um conjunto de testes, a fase de desenvolvimento pode começar. Usando uma abordagem MDSD, o desenvolvedor cria ou amplia os modelos existentes gerando uma maior versão da aplicação. A geração de código é realizado automaticamente por meio da transformação que toma como entrada os modelos.
- Etapa 4: A fim de verificar se o requisito foi implementado com sucesso e nenhuma funcionalidade anterior está corrompida, o programador executa todo o conjunto de testes para verificar ambas as coisas. Se um ou mais testes falharem, ele deve voltar para a etapa 3, fazer algum retrabalho nos modelos, gerar o código novamente, e repetir a etapa 4 até que todos os testes passem.
- Etapa 5: Finalmente, tem-se uma nova aplicação com um requisito adicional. O ciclo continua com a escolha de um novo requisito (Etapa 1) e seguindo as etapas de 2 a 5 até que termine os requisitos do *sprint*.

### 3.3.3 Avaliação de Usabilidade em Diagramas de Atividades

Antes da criação da abordagem para melhorar o apoio de ferramenta de usabilidade automatizada durante o desenvolvimento de *websites*, ATTERER (2005) realizou um estudo de caso onde, dentre outros aspectos, buscou-se examinar modelos de apresentação e navegação quanto à possibilidade de expressar constrangimentos de usabilidade de duas abordagens de engenharia *web*: UWE (*UML-based Web Engineering*, KOCH e KRAUS 2002)

e OO-H (*Objectoriented Hypermedia*, GÓMEZ *et al.* 2001). A abordagem foi seguida criando-se diagramas de atividade da UML para o exemplo e através deste uma série de notas sobre a usabilidade do processo pôde ser feita. No entanto, foi identificado que o modelo (diagrama de atividade) era abstrato demais para ser convertido em páginas *web*. Para soluções de Engenharia *Web* com suporte à ferramenta, as informações para gerar automaticamente a estrutura de *links* e conteúdo da página estão disponíveis nos modelos de navegação e apresentação, porém nenhum desses modelos foi criado para o protótipo. Em vez disso, o modelo de atividade serviu como base para a criação *ad hoc* manual de adequados *links* e conteúdo da página. Segundos os autores, na prática, isso funcionou muito bem, uma vez que apenas um modelo está presente, você não precisa alternar entre aos pontos de vista da “lógica da aplicação”, “navegação” e “apresentação” da aplicação *web*, e os diagramas diferentes para estes pontos de vista não precisam ser mantidos em sincronia. Além disso, os autores argumentam que normalmente é fácil chegar a um conteúdo da página adequado por um determinado modelo. Após este estudo, ATTERER (2008) criou uma abordagem que usa modelos abstratos de abordagens de engenharia *web* durante a validação automática de usabilidade.

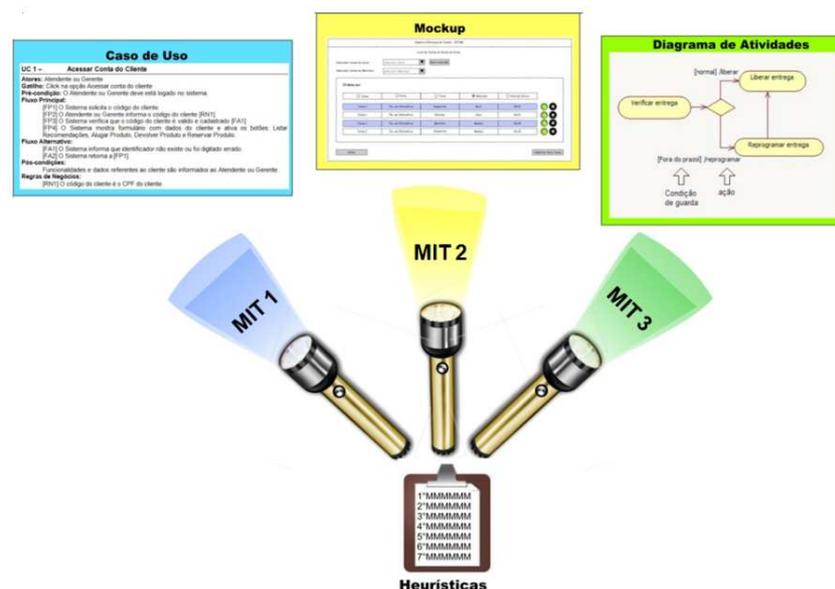
Já SILVA e SILVEIRA (2008) realizaram um estudo experimental para validar a técnica de avaliação de usabilidade em diagramas de atividades, onde se buscou verificar a correlação entre os resultados da aplicação do método proposto sobre o modelo e do mesmo método sobre o sistema já desenvolvido, a fim de descobrir quantos problemas poderiam ser identificados no diagrama de atividades. No entanto, esta validação deu-se apenas pela aplicação do SMC (*Simple Matching Coefficient*), o que não permitia uma generalização dos resultados. Com isso, SILVA e SILVEIRA (2010) executaram outro estudo, onde optaram por avaliar a usabilidade do sistema, aplicando as diretrizes propostas sobre o diagrama de atividades e então avaliar o mesmo sistema – já desenvolvido – realizando uma avaliação heurística. Neste segundo estudo, os dados experimentais foram interpretados através de análises quantitativas e qualitativas, e o resultado apontou que a técnica ajudou a encontrar problemas de usabilidade neste modelo.

### 3.4 Proposta do Conjunto de Técnicas MIT

Esta Subseção apresenta a proposta desta dissertação, o conjunto de técnicas MIT (*Model Inspection Technique for Usability Evaluation*), que são técnicas de leitura para inspeção de usabilidade em modelos de análise e projeto. Segundo TRAVASSOS *et al.* (2002), técnicas de leitura são um tipo específico de técnica de inspeção que contém uma série de

passos para a análise individual de um produto de *software* de forma a alcançar a compreensão necessária para uma tarefa específica. Estas têm como objetivo aumentar a eficácia das inspeções, fornecendo diretrizes que podem ser utilizados pelos revisores para analisar um determinado artefato de *software* e identificar defeitos (TRAVASSOS *et al.* 2002). Desta forma, as técnicas propostas possuem um conjunto de itens de verificação que devem guiar o inspetor durante a avaliação de usabilidade em Caso de Uso, em *Mockups* e em Diagrama de Atividades (Figura 2).

Para a realização da inspeção de usabilidade através destas técnicas é sugerido haver mais que um avaliador, pois, segundo ROCHA E BARANAUSKAS (2003), uma única pessoa nunca é capaz de encontrar todos os problemas de usabilidade.



**Figura 2** - Visão geral do Conjunto de Técnicas MIT

A MIT 1 leva em conta a avaliação de usabilidade em Casos de Uso presente no método UCE (HORNBAEK *et al.* 2007). No entanto, este método, como dito anteriormente, é composto por três atividades, o que torna o processo de inspeção demorado. O objetivo dessa customização é melhorar a produtividade de aplicação da inspeção. Através das heurísticas de Nielsen citadas no UCE e com o acréscimo de outras heurísticas foram criados os itens de verificação. A ideia de criar estes itens surgiu com objetivo de gerar um maior nível de detalhamento, orientando o inspetor a encontrar os possíveis defeitos de usabilidade nos Casos de Uso. A versão inicial da MIT 1 possui 21 itens de verificação, agrupados nas seguintes heurísticas: Visibilidade do status do sistema (Figura 3); Concordância entre o sistema e o mundo real; Controle e liberdade do usuário; Consistência e padrões; Prevenção de erros; Reconhecer ao invés de lembrar; Flexibilidade

e eficiência de uso; Reconhecimento, diagnóstico e recuperação de erros; e Proporcionar funções que são úteis para o usuário.

<b>MIT-1A. Visibilidade do Status do Sistema</b>	
<b>MIT-1A1</b>	Verifique se há algum texto no FP, FA e FE que informa em que parte do sistema o usuário se encontra;
<b>MIT-1A2</b>	Verifique se há algum texto no FP, FA e FE que informa ao usuário o que foi realizado após uma persistência de dados. Por exemplo: quando há alteração ou exclusão de algo, uma mensagem de texto é apresentada.

**Figura 3** - Itens de verificação da heurística Visibilidade do status do Sistema (MIT 1 v1)

Já a MIT 2 considera a avaliação de usabilidade em *mockups* presente na abordagem de LUNA *et al.* (2010). Porém, a proposta de LUNA *et al.* (2010) utiliza testes de interação executados em ferramenta própria, o que não permite seu uso sem esta ferramenta. A MIT 2 utiliza procedimentos de leitura para os inspetores avaliarem os *mockups* através de 32 itens de verificação, agrupados nas seguintes heurísticas: Visibilidade do status do sistema; Concordância entre o sistema e o mundo real; Controle e liberdade do usuário (Figura 4); Consistência e padrões; Prevenção de erros; Reconhecer ao invés de lembrar; Flexibilidade e eficiência de uso; Reconhecimento, diagnóstico e recuperação de erros; e proporcionar funções que são úteis para o usuário.

<b>MIT-2C. Controle e liberdade ao usuário</b>	
<b>MIT-2C1</b>	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;
<b>MIT-2C2</b>	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado. Por exemplo, ter a possibilidade de clicar na opção “Cancelar” ou “Sair”;

**Figura 4** - Itens de verificação da heurística Controle e liberdade ao usuário (MIT 2 v1)

Por fim, a MIT 3 faz avaliação de usabilidade em diagrama de atividades, conforme sugerido em um estudo inicial da abordagem proposta por ATTERER (2008) e a abordagem de SILVA e SILVEIRA (2010). Deve-se notar que a técnica proposta por ATTERER (2008) não realiza inspeção em diagramas de atividades, este diagrama serviu apenas como base para a criação manual de *links* adequados e conteúdo de página, para posterior criação de modelos de navegação. Porém, devido ao uso do diagrama de atividades para a descrição das interações nas funcionalidades, decidiu-se selecioná-lo como modelo a ser inspecionado. A abordagem de SILVA e SILVEIRA (2010) possui foco em avaliação de usabilidade de diagrama de atividades, porém a técnica não está disponível para uso, impossibilitando assim uma possível análise de suas heurísticas. Portanto, esta técnica também contribuiu apenas com a ideia de avaliação de usabilidade em diagrama de atividades. A versão inicial da MIT 3 possui 13 itens de verificação, agrupados nas seguintes

heurísticas: Concordância entre o sistema e o mundo real (Figura 5); Controle e liberdade do usuário; Consistência e padrões; Prevenção de erros; Reconhecer ao invés de lembrar; Flexibilidade e eficiência de uso; Reconhecimento, diagnóstico e recuperação de erros; e proporcionar funções que são úteis para o usuário.

MIT-3A. Concordância entre o sistema e o mundo real	
<b>MIT-3A1</b>	Verifique se as atividades estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema;
<b>MIT-3A2</b>	Verifique se os nomes das atividades utilizam termos (palavras) que seguem as convenções do mundo real, ou seja, que englobam tanto as convenções do domínio do problema quanto às convenções de terminologia de aplicações semelhantes.

Figura 5 - Itens de verificação da heurística Concordância entre o sistema e o mundo real (MIT 3 v1)

A Figura 6 apresenta o conjunto de técnicas MIT e as abordagens que contribuíram com ideias e heurísticas para cada MIT.

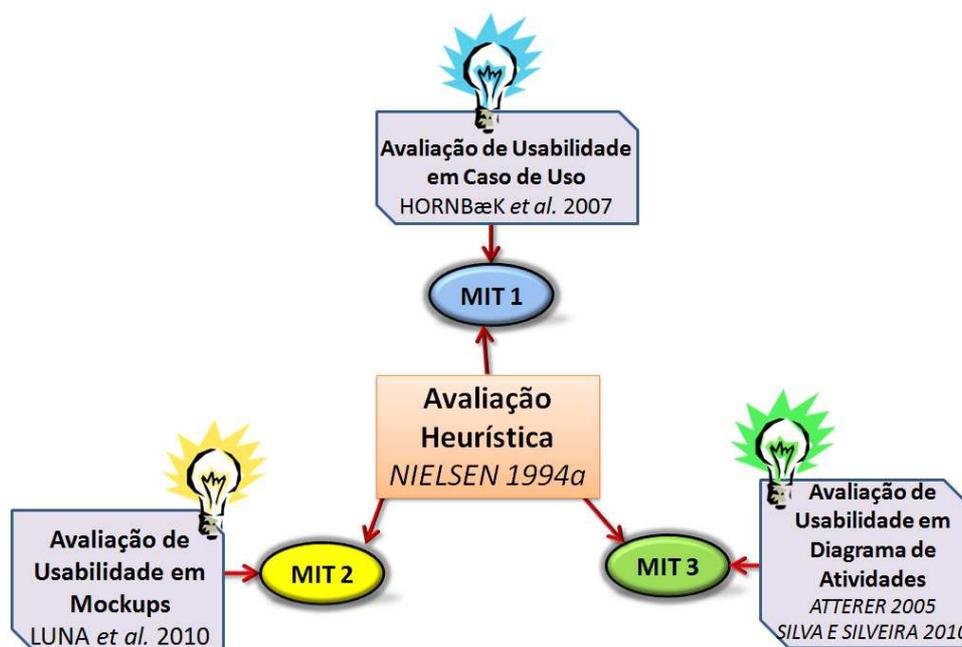


Figura 6 - Formação do Conjunto de Técnicas MIT

O texto completo da primeira versão das MITs (MIT v1) é apresentado na Tabela 11 a seguir.

Tabela 11 - Primeira versão das MITs

MIT 1 – Avaliação de Usabilidade em Casos de Uso	
<b>Objetivo:</b>	Verificar a usabilidade em Casos de Uso
<b>Instruções:</b>	As instruções para avaliação da MIT 1 são: 1. Antes de começar a inspeção, observe as seguintes diretrizes: a. Toda mensagem do sistema (mensagens de erro, textos informativos,

	<p>advertências) e todo nome de botões, <i>links</i>, opções, entre outros, deve estar claramente descrito para o usuário no Caso de Uso.</p> <p>b. Para toda ação importante do usuário no Casos de Uso (ações que envolvam persistência de dados, alteração de estado de sistema, entre outros), o sistema deve perguntar pela confirmação de sua ação.</p> <p>2. Comece a inspeção. Para cada Caso de Uso a ser inspecionado observe e anote em qual fluxo principal, alternativo, de exceção ou regra de negócio você identifica um problema de usabilidade seguindo os itens de verificação abaixo:</p>
<b>Itens de Verificação - MIT 1</b>	
<b>MIT-1A. Visibilidade do Status do Sistema</b>	
<b>MIT-1A1</b>	Verifique se há algum texto no FP <sup>3</sup> , FA <sup>4</sup> e FE <sup>5</sup> que informa em que parte do sistema o usuário se encontra;
<b>MIT-1A2</b>	Verifique se há algum texto no FP, FA e FE que informa ao usuário o que foi realizado após uma persistência de dados. Por exemplo: quando há alteração ou exclusão de algo, uma mensagem de texto é apresentada.
<b>MIT-1B. Concordância entre o sistema e o mundo real</b>	
<b>MIT-1B1</b>	Verifique se os nomes de campos, telas, botões, links, mensagens de erros e textos informativos no FP, FA, FE e RN <sup>6</sup> apresentam conceitos familiares aos usuários, ou seja, segue as convenções do mundo real;
<b>MIT-1B2</b>	Verifique se as opções ou campos informados pelo sistema no FP, FA e FE estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>MIT-1C. Controle e liberdade ao usuário</b>	
<b>MIT-1C1</b>	Verifique se o usuário, através do FA e FE, pode desfazer ou refazer algo que envolva persistência de dados no sistema. Por exemplo: pode excluir ou alterar dados inseridos.
<b>MIT-1D. Consistência e padrões</b>	
<b>MIT-1D1</b>	Verifique se todos os botões ou links possuem nomes que indicam funcionalidades diferentes no FP, FA, FE e RN, ou seja, não há botões ou links com nomes diferentes, mas com a mesma funcionalidade;
<b>MIT-1D2</b>	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões apresentados no FP, FA e FE.
<b>MIT-1E. Prevenção de Erros</b>	
<b>MIT-1E1</b>	Verifique se há alguma advertência do sistema que alerta o usuário através de mensagens ou textos informativos de que o que ele está fazendo pode ser inapropriado naquele momento no FP, FA, FE;

<sup>3</sup> FP: Fluxo Principal

<sup>4</sup> FA: Fluxo Alternativo

<sup>5</sup> FE: Fluxo Exceção

<sup>6</sup> RN: Regras de Negócio

<b>MIT-1E2</b>	Verifique se todas as opções, botões e links disponíveis possuem nomes que definem claramente que resultados ou estados serão atingidos no FP <sup>7</sup> , FA <sup>8</sup> , FE <sup>9</sup> e RN <sup>10</sup> .
<b>MIT-1F. Reconhecer ao invés de lembrar</b>	
<b>MIT-1F1</b>	Verifique se os nomes das opções, campos, telas e links são informados de forma que o usuário não tenha que se lembrar quais são eles no FP, FA, FE e RN.
<b>MIT-1G. Flexibilidade e eficiência de uso</b>	
<b>MIT-1G1</b>	Verifique se há nomes de links, nome de telas ou nome de campos abreviados para usuários mais experientes no FP, FA, FE e RN;
<b>MIT-1G2</b>	Verifique se há como passos serem realizados de forma mais rápida e eficiente para usuários mais experientes no FP, FA, FE e RN;
<b>MIT-1G3</b>	Verifique se o sistema permite que o usuário realize o que ele deseja com facilidade através dos passos do FA e FE;
<b>MIT-1G4</b>	Verifique se o sistema permite ao usuário formas de acesso às tarefas principais no FA e FE. Por exemplo: estando no FA ou FE o usuário consegue voltar ao FP quando for viável;
<b>MIT-1G5</b>	Verifique se as formas de acesso oferecidas pelo sistema no FA e FE diminuem o esforço de ações físicas dos usuários.
<b>MIT-1H. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>MIT-1H1</b>	Verifique se as mensagens de erros utilizam linguagem simples de ser entendida pelos usuários, ou seja, de acordo com o perfil do usuário no FP, FA e FE. Por exemplo: mensagens de erro que não tenha códigos;
<b>MIT-1H2</b>	Verifique se todas as mensagens de erros não culpam ou intimidam o usuário no FP, FA, FE;
<b>MIT-1H3</b>	Verifique se as mensagens de erro ajudam o usuário a corrigir o erro no FP, FA e FE. Por exemplo: mensagens com indicação de recuperação;
<b>MIT-1H4</b>	Verifique se o sistema mostra como acessar as soluções alternativas que são apresentadas em algumas mensagens de erro no FP, FA e FE.
<b>MIT-1I. Proporcionar funções que são úteis para o usuário</b>	
<b>MIT-1I1</b>	Verifique se todo botão ou link essencial está presente no FP, FA, FE e RN;
<b>MIT-1I2</b>	Verifique se todo passo, botão ou link presente é útil no FP, FA, FE e RN.
<b>MIT 2 – Avaliação de Usabilidade em <i>Mockups</i></b>	
<b>Objetivo:</b>	Verificar a usabilidade em <i>Mockups</i>
<b>Instruções:</b>	As instruções para avaliação da MIT 2 são:

<sup>7</sup> FP: Fluxo Principal

<sup>8</sup> FA: Fluxo Alternativo

<sup>9</sup> FE: Fluxo Exceção

<sup>10</sup> RN: Regras de Negócio

	<p>1. Antes de começar a inspeção, observe a seguinte diretriz:</p> <p>a. Toda mensagem do sistema (mensagens de erro, textos informativos, advertências) e todo nome de botões, <i>links</i>, opções, entre outros, deve estar claramente visível para o usuário nos <i>Mockups</i>.</p> <p>2. Comece a inspeção. Para cada <i>Mockups</i> a ser inspecionado observe e anote qual problema de usabilidade você encontrou seguindo os itens de verificação a seguir:</p>
<b>Itens de Verificação - MIT 2</b>	
<b>MIT-2A. Visibilidade do Status do Sistema</b>	
<b>MIT-2A1</b>	Verifique se há informações textuais ou nome nos <i>Mockups</i> que informa em que parte do sistema o usuário se encontra.
<b>MIT-2A2</b>	Verifique se há algum texto informativo ou mensagem que informa ao usuário o que foi realizado após uma persistência de dados. Por exemplo: quando há alteração ou exclusão de algo, uma mensagem de texto é apresentada.
<b>MIT-2B. Concordância entre o sistema e o mundo real</b>	
<b>MIT-2B1</b>	Verifique se os símbolos, palavras, frases, nome de botões, links, mensagens de erros e textos informativos estão expressas em uma representação facilmente compreendida pelo usuário, ou seja, são baseadas em uma linguagem familiar aos usuários;
<b>MIT-2B2</b>	Verifique se as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>MIT-2C. Controle e liberdade ao usuário</b>	
<b>MIT-2C1</b>	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;
<b>MIT-2C2</b>	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado. Por exemplo, ter a possibilidade de clicar na opção “Cancelar” ou “Sair”;
<b>MIT-2D. Consistência e padrões</b>	
<b>MIT-2D1</b>	Verifique se a terminologia, gráficos e símbolos estão consistentes;
<b>MIT-2D2</b>	Verifique se há padrões adotados em relação a layout, formatação e controles. Por exemplo, botão “Salvar” no canto superior direito em um <i>Mockup</i> e em outro no canto inferior direito;
<b>MIT-2D3</b>	Verifique se todos os botões/links possuem nomes/ícones que indicam funcionalidades diferentes, ou seja, não há botões/links com nomes/ícones diferentes, mas com a mesma funcionalidade;
<b>MIT-2D4</b>	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões.
<b>MIT-2E. Prevenção de Erros</b>	
<b>MIT-2E1</b>	Verifique se os dados obrigatórios na entrada de dados estão claramente definidos;

<b>MIT-2E2</b>	Verifique se há alguma indicação do formato correto para uma entrada de dados específica;
<b>MIT-2E3</b>	Verifique se há alguma advertência do sistema que alerta o usuário através de mensagens ou textos informativos de que o que ele está fazendo pode ser inapropriado naquele momento;
<b>MIT-2E4</b>	Verifique se todas as opções, botões e links disponíveis possuem nomes que definem claramente que resultados ou estados serão atingidos.
<b>MIT-2F. Reconhecer ao invés de lembrar</b>	
<b>MIT-2F1</b>	Verifique se os nomes das opções, campos, botões e links são informados de forma que o usuário não tenha que se lembrar quais são eles;
<b>MIT-2F2</b>	Verifique se os ícones/figuras minimizam ou reduzem a carga cognitiva e o esforço físico do usuário de lembrar;
<b>MIT-2F3</b>	Verifique se é fácil reconhecer/visualizar a opção que deve ser usada para atingir o objetivo desejado;
<b>MIT-2F4</b>	Verifique se é fácil reconhecer/visualizar dados já fornecidos.
<b>MIT-2G. Flexibilidade e eficiência de uso</b>	
<b>MIT-2G1</b>	Verifique se a disposição dos elementos nos <i>Mockups</i> aumenta a eficiência de uso, minimizando o esforço de ações físicas;
<b>MIT-2G2</b>	Verifique se há apoio a tarefas específicas frequentemente repetidas;
<b>MIT-2G3</b>	Verifique se há diferentes formas de acesso às tarefas principais, ou seja, possibilidade de utilização de aceleradores ou atalhos;
<b>MIT-2G4</b>	Verifique se há facilidade para entrada de dados, sejam eles simples ou complexos estruturalmente;
<b>MIT-2G5</b>	Verifique se há mecanismos de busca de informação que auxiliem a entrada de dados obrigatória;
<b>MIT-2G6</b>	Verifique se há nomes de links abreviados para usuários mais experientes;
<b>MIT-2G7</b>	Verifique se há como as funcionalidades do sistema serem realizadas de forma mais rápida e eficiente para usuários mais experientes.
<b>MIT-2H. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>MIT-2H1</b>	Verifique se as mensagens de erros estão claramente visíveis para o usuário;
<b>MIT-2H2</b>	Verifique se as mensagens de erros utilizam linguagem simples de ser entendida pelos usuários, ou seja, de acordo com o perfil do usuário. Por exemplo: mensagens de erro que não tenha códigos;
<b>MIT-2H3</b>	Verifique se todas as mensagens de erros não culpam ou intimidam o usuário;
<b>MIT-2H4</b>	Verifique se as mensagens de erro ajudam o usuário a corrigir o erro. Por exemplo: mensagens com indicação de recuperação;
<b>MIT-2H5</b>	Verifique se o sistema mostra como acessar as soluções alternativas que são apresentadas em algumas nas mensagens de erro.
<b>MIT-2I. Proporcionar funções que são úteis para o usuário</b>	
<b>MIT-2I1</b>	Verifique se toda funcionalidade do sistema, botão ou link essencial está

	presente;
<b>MIT-2I2</b>	Verifique se toda funcionalidade do sistema, botão ou link presente é útil.
<b>MIT 3 – Avaliação de Usabilidade em Diagrama de Atividades</b>	
<b>Objetivo:</b>	Verificar a usabilidade em Diagrama de Atividades
<b>Instruções:</b>	As instruções para avaliação da MIT 3 são: 1. Antes de começar a inspeção, observe as seguintes diretrizes: a. Toda atividade, transição, condição de guarda ou sinal (mensagem enviada ou recebida) do Diagrama de Atividade deve estar descrita no Caso de Uso. 2. Comece a inspeção. Para cada Diagrama de Atividades a ser inspecionado observe e anote qual problema de usabilidade você encontrou seguindo os itens de verificação abaixo:
<b>Itens de Verificação - MIT 3</b>	
<b>MIT-3A. Concordância entre o sistema e o mundo real</b>	
<b>MIT-3A1</b>	Verifique se as atividades estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema;
<b>MIT-3A2</b>	Verifique se os nomes das atividades utilizam termos (palavras) que seguem as convenções do mundo real, ou seja, que englobam tanto as convenções do domínio do problema quanto às convenções de terminologia de aplicações semelhantes.
<b>MIT-3B. Controle e liberdade ao usuário</b>	
<b>MIT-3B1</b>	Verifique se o usuário através de alguma atividade consegue alterar dados que ele inseriu através de outra atividade;
<b>MIT-3B2</b>	Verifique se o usuário através de alguma atividade consegue excluir dados que ele inseriu através de outra atividade.
<b>MIT-3C. Consistência e padrões</b>	
<b>MIT-3C1</b>	Verifique se todas as atividades indicam funcionalidades diferentes, ou seja, não há atividades com nomes diferentes, mas com a mesma funcionalidade.
<b>MIT-3D. Prevenção de Erros</b>	
<b>MIT-3D1</b>	Verifique se há alguma atividade em que o sistema previne algum tipo de erro.
<b>MIT-3E. Reconhecer ao invés de lembrar</b>	
<b>MIT-3E1</b>	Verifique se é fácil reconhecer a atividade que deve ser usada para atingir o objetivo desejado.
<b>MIT-3F. Flexibilidade e eficiência de uso</b>	
<b>MIT-3F1</b>	Verifique se o usuário possui diferentes formas de acesso às atividades principais;
<b>MIT-3F2</b>	Verifique se o usuário pode navegar com facilidade pelas diferentes atividades, ou seja, as possibilidades de navegação para atingir um objetivo estão claras para o usuário.

<b>MIT-3G. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>MIT-3G1</b>	Verifique através das atividades se o sistema ajuda o usuário a sair de uma situação de erro;
<b>MIT-3G2</b>	Verifique através das atividades se o sistema ajuda a corrigir o erro. Por exemplo: atividades que indicam a recuperação do erro.
<b>MIT-3H. Proporcionar funções que são úteis para o usuário</b>	
<b>MIT-3H1</b>	Verifique se todas as atividades são úteis;
<b>MIT-3H2</b>	Verifique se todas as atividades úteis estão presentes.

A versão completa inicial do conjunto de técnicas MIT também está descrita detalhadamente em um Relatório Técnico (VALENTIM e CONTE 2012). O propósito deste conjunto de técnicas é que seja facilmente adotável pela indústria de desenvolvimento de *software*. Além disso, outros objetivos são diminuir o esforço de retrabalho e reduzir o custo das correções dos problemas de usabilidade, já que é mais barato resolver estes problemas em estágios iniciais de desenvolvimento do que quando o sistema se encontra finalizado.

Uma das características das MITs é que elas não necessitam de apoio ferramental. Outra característica relevante é que as técnicas podem ser usadas de forma independente. No entanto, melhores resultados são encontrados se utilizadas em conjunto pelo fato de integrar avaliação de artefatos de ES e IHC com o mesmo propósito de avaliação de usabilidade.

Após a análise das técnicas encontradas no Mapeamento Sistemático (presente na Subseção 3.2) e a proposta das MITs (presente na atual subseção), observou-se que as MITs poderiam inspecionar a usabilidade tanto de modelos usados no processo de desenvolvimento de aplicações *Web* quanto em modelos usados no processo de desenvolvimento de sistemas em geral. Fazendo com que as MITs possam garantir a usabilidade abrangendo uma maior quantidade de modelos.

### **3.5 Considerações Finais**

Este capítulo teve como objetivo apresentar os métodos de avaliações de usabilidade bases para esta proposta. Adicionalmente foi descrita a técnica proposta formada pelo conjunto de três técnicas: Avaliação dos Casos de Uso, Avaliação dos *Mockups* e Avaliação do Diagrama de Atividades.

O próximo capítulo descreve um estudo primário executado de acordo com a metodologia apresentada no Capítulo 1, utilizado para a avaliação e aprimoramento do conjunto de técnicas MIT.

# CAPÍTULO 4 - AVALIAÇÃO DO CONJUNTO DE TÉCNICAS MIT ATRAVÉS DE UM ESTUDO EXPERIMENTAL

## 4.1 Introdução

Segundo PFLEEGER (1999), nenhuma ciência pode avançar sem experimentação e medição. E de acordo com SHULL *et al.* (2001), estudos experimentais devem ser realizados e repetidos para provar a veracidade da pesquisa, deixando a outros pesquisadores o conhecimento usado na execução de um experimento e possibilitando um melhor entendimento e análise do estudo realizado. A meta com a execução e repetição dos estudos experimentais é construir um corpo de conhecimento baseado em experimentação que identifica as vantagens e os custos das diferentes técnicas e ferramentas de apoio à Engenharia de *Software* (SHULL *et al.* 2004). Esta pesquisa está apoiada no uso da experimentação. Desta forma, este capítulo apresenta o primeiro estudo de viabilidade executado, conforme proposto na metodologia. A Subseção 4.2 apresenta o planejamento e execução deste estudo, a análise quantitativa e qualitativa, além das ameaças a validade. E por fim a Subseção 4.3 apresenta as conclusões

## 4.2 Estudo de Viabilidade

Para SHULL *et al.* (2001), o primeiro estudo que se deve realizar para avaliar uma nova tecnologia é um estudo de viabilidade, que tem por objetivo verificar se esta nova tecnologia é viável e se o tempo empregado é bem utilizado. Estes estudos podem ser realizados em ambiente acadêmico, pois isso permite que as novas tecnologias sejam testadas antes de serem transferidas para indústria usá-las de forma segura em seus processos.

O estudo de viabilidade desta pesquisa teve como objetivo avaliar o conjunto de técnicas MIT em relação à AH considerando que esta foi a técnica utilizada como base para sua definição (VALENTIM *et al.* 2012b).

### 4.2.1 Planejamento do Estudo de Viabilidade

O Caso de Uso e os *Mockups* avaliados neste experimento fazem parte da especificação de um sistema real para Gerenciamento dos Cursos e Turmas de um Centro de Treinamento. E o Diagrama de Atividades foi elaborado a partir do Caso de Uso deste

Centro de Treinamento. Foram inseridos 11 defeitos no Caso de Uso, 6 defeitos nos *Mockups* e 6 defeitos no Diagrama de Atividades. A Figura 7 apresenta um exemplo de defeito inserido no Caso de Uso, onde a mensagem em destaque exibe um texto em caixa alta. O usuário ao se deparar com esta mensagem pode se sentir intimidado, já que da forma que está escrito parece que o sistema está culpando o usuário por um erro. Além disso, a mensagem em destaque não é apresentada de acordo com o padrão das outras mensagens. E por fim, a mensagem não ajuda o usuário a corrigir o erro, pois não mostra que campo está vazio.

**Fluxo de Exceção:**

**E1.- O Funcionário cadastra um Curso já cadastrado.**

- E1.1.- O funcionário insere os dados de um Curso já existente.
- E1.2.- O sistema mostra mensagem: "O curso já existe".
- E1.3.- O caso de uso é encerrado.

**E2.- O funcionário deixa de inserir dados obrigatórios.**

- E2.1.- Se o funcionário deixa de inserir dados no campo **Nome**.
  - 2.1.1. O sistema mostra mensagem: "Campo vazio".
- E2.2.- Se o funcionário deixa de selecionar uma das opções do campo **Segmento**.
  - 2.2.1. O sistema mostra mensagem: "Campo Segmento está vazio".
- E2.3.- Se o funcionário deixa de selecionar uma das opções do campo **Eixo Tecnológico** (Para Segmento = Formação Técnica).
  - 2.3.1. O sistema mostra mensagem: "CAMPO VAZIO! PREENCHA!".
- E2.4.- Se o funcionário deixa de selecionar uma das opções do campo **Tipo** (Para Segmento = Formação Técnica).
  - 2.4.1. O sistema mostra mensagem: "Campo Tipo não contém nenhum dado".
- E2.5.- O caso de uso é encerrado.

Figura 7 – Exemplo de defeito inserido no Caso de Uso

A Figura 8 apresenta um exemplo de defeito inserido no *Mockup*, onde os campos em destaque não indicam o formato correto para a entrada de dados. Ou seja, o usuário não sabe se a Data de Início e da Data de Término é no formato DD/MM/AAAA ou DD/MM/AA.

Cadastro de Turmas - Centro de Treinamento

Curso\*

Localizacao\*  Capital  Interior

Município\*

Status

Data de Início\*

Data de Término\*

Programa de Trabalho\*

Turno\*  Matutino (Manha)  Vespertino (Tarde)  Noturno (Noite)

Unidade ou Centro de Formação

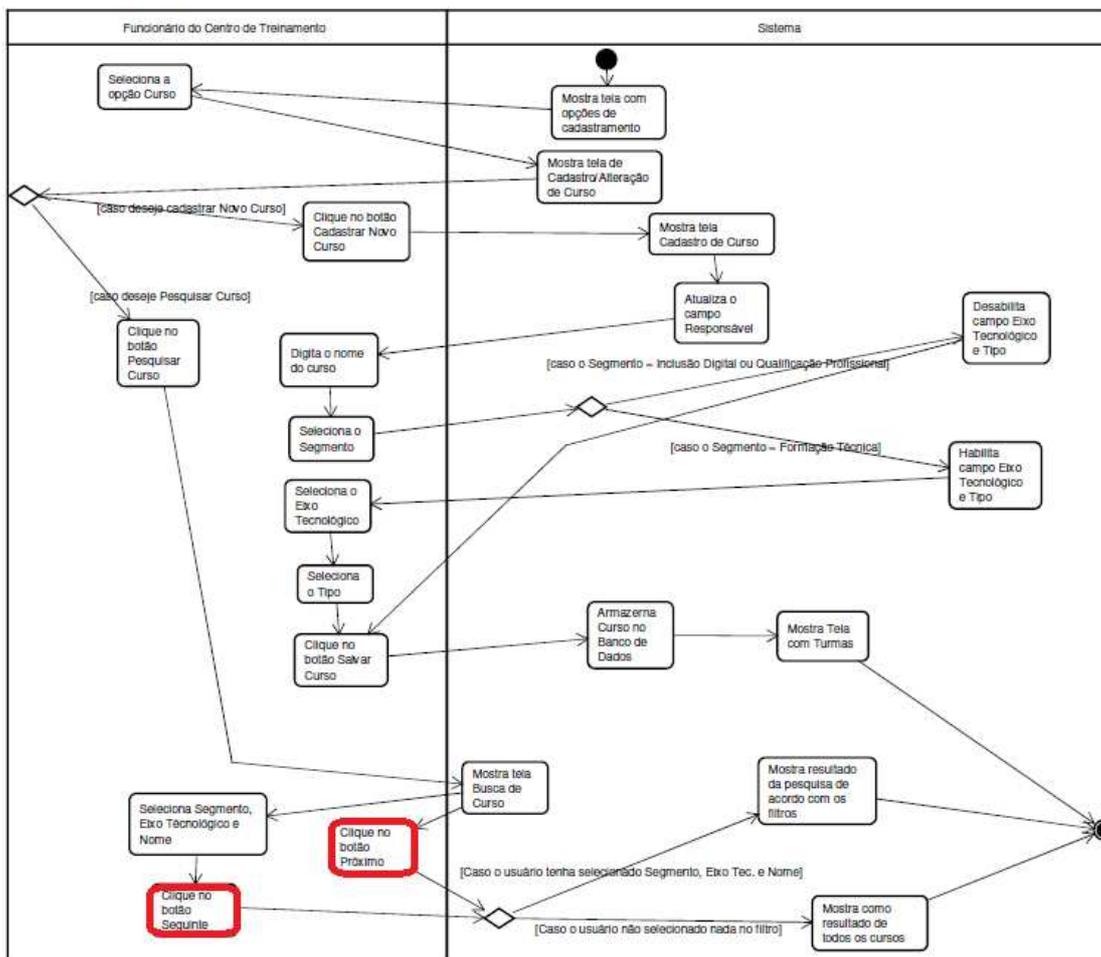
Instituição(aes) Parceira(s)

Vigência TCT

Pesquisar

Figura 8 – Exemplo de defeito inserido no *Mockup*

A Figura 9 apresenta um exemplo de defeito inserido no Diagrama de Atividades, onde as atividades em destaque possuem a mesma funcionalidade, porém com nomes diferentes.



**Figura 9** – Exemplo de defeito inserido no Diagrama de Atividades

Os participantes deste estudo foram alunos voluntários da disciplina Modelagem e Projeto de Sistemas (MPS) do 5º período do curso de graduação em Sistemas de Informação (SI) da Universidade Federal do Amazonas (UFAM). Vale ressaltar que a disciplina MPS é focada em modelagem de projetos e que os alunos participantes deste estudo já haviam tido uma disciplina introdutória sobre modelagem no período anterior. Além disso, vários destes alunos possuem experiência em desenvolvimento na indústria.

Ao todo, 16 alunos concordaram em participar do estudo. Todos os participantes assinaram um Termo de Consentimento Livre e Esclarecido (TCLE) e preencheram um Formulário de Caracterização que continha perguntas sobre o conhecimento dos participantes em usabilidade e sua experiência em desenvolvimento de *software*.

O atributo de caracterização do conhecimento em usabilidade foi classificado de acordo com a seguinte escala:

- Nenhum conhecimento: não conhece nada sobre usabilidade;
- Baixo conhecimento: noções de usabilidade adquirida através de leituras/palestras;
- Médio conhecimento: participação em projetos ou avaliações de usabilidade na sala de aula;
- Alto conhecimento: participação em projetos ou avaliações de usabilidade na indústria.

Já o atributo experiência em desenvolvimento de *software* foi classificado em:

- Baixa experiência: menos de 1 ano de experiência;
- Média experiência: de 1 a 4 anos de experiência;
- Alta experiência: mais de 4 anos de experiência.

Dos 16 alunos, um participante caracterizado para o grupo da AH faltou no dia do experimento. Desta forma, os 15 participantes foram divididos em dois grupos, de forma a distribuir igualmente os participantes com conhecimento em usabilidade e experiência em desenvolvimento. Um dos grupos usou as técnicas MIT e o outro grupo usou a AH para a inspeção de usabilidade de um Caso de Uso, dois *Mockups* e um Diagrama de Atividades. A Tabela 12 apresenta a caracterização da experiência dos inspetores e seus respectivos grupos.

**Tabela 12** - Caracterização dos Inspetores

Técnica/Inspetor		Conhecimento em Usabilidade	Experiência em Desenvolvimento
MIT	1	Nenhum	Baixo
	2	Baixo	Baixo
	3	Baixo	Médio
	4	Nenhum	Baixo
	5	Médio	Alto
	6	Nenhum	Baixo
	7	Médio	Médio
	8	Nenhum	Médio
AH	9	Nenhum	Médio
	10	Baixo	Baixo
	11	Médio	Baixo
	12	Alto	Alto
	13	Baixo	Médio
	14	Nenhum	Médio
	15	Nenhum	Baixo

Uma apresentação sobre usabilidade foi realizada para todos os participantes do estudo. E para o grupo que usaria o conjunto de técnicas MIT, foi realizada uma breve apresentação sobre as técnicas MIT.

#### 4.2.2 Execução da Inspeção de Usabilidade

Ao início da execução, uma pesquisadora atuou como moderadora, sendo responsável por passar as informações para execução da avaliação aos inspetores. No primeiro dia de inspeção, uma das orientações dadas foi que primeiro seria feita a inspeção em casos de uso, depois em *Mockups*. No segundo dia, seria feita a inspeção em Diagramas de Atividades.

Em seguida os grupos foram divididos e ficaram em salas diferentes. Cada participante recebeu as Instruções para Inspeção, a técnica de inspeção (conjunto de técnicas MIT para o grupo que usaria as MIT's ou documento com a AH para o grupo que usaria a avaliação heurística), planilha para a anotação das discrepâncias encontradas (Figura 10), os modelos (Caso de Uso, *Mockups* e Diagrama de Atividades) a serem inspecionados, além de um questionário de avaliação pós-inspeção, apresentado no Apêndice A.

**Planilha de Discrepâncias nos Casos de Uso usando a MIT**

Inspetor:

Tempo de inspeção:  minutos

Nro	Heurística	Nro do Caso de Uso	Passo do Caso de Uso	Descrição do Problema Encontrado
1				
2				
3				
4				

**Figura 10** – Exemplo da planilha de anotação de discrepâncias

Ao todo, participaram 8 inspetores usando o conjunto de técnicas MIT's e 7 inspetores usando a AH. A atividade de detecção de defeitos foi realizada individualmente pelos inspetores e vale ressaltar que durante esta atividade os inspetores não receberam nenhum auxílio dos pesquisadores envolvidos no estudo. As Figuras 11, 12 e 13 a seguir mostram respectivamente exemplos de defeitos anotados nas planilhas de discrepâncias pelo inspetor durante a avaliação de usabilidade no Caso de Uso, no *Mockup* e no Diagrama de Atividades, sendo que os defeitos em destaques são os mesmos evidenciados nas Figuras 7, 8 e 9 anteriormente apresentadas.

Inspetor:	XXXXXXXX			
Tempo de inspeção:	100 minutos			
Nro	Heurística	Nro do Caso de Uso	Passo do Caso de Uso	Descrição do Problema Encontrado
1	MIT - 1G5	01	FP3	Quando o usuário clica em "Cursos", o sistema exibe a tela de cadastro/alteração de cursos, onde tem os botões Cadastrar/Alterar e Pesquisar. Ficaria melhor se fossem listados todos os cursos e mostradas as opções, pois já seria possível deletar ou alterar um curso, sem necessariamente ter que pesquisar.
2	MIT - 1H1	01	FP3	Não são devidamente informados no FP quais são as ações disponíveis para o usuário, sendo possível descobri-las apenas através dos FA's.
3		01	FP6	Não são devidamente informados quais são os campos presentes no formulário de cadastro de curso.
4	MIT - 1E1	01	FP7	Não são indicados quais são os campos obrigatórios no formulário de cadastro/alteração de curso.
5	MIT - 1A2	01	FP9	Não é apresentada uma mensagem de confirmação para cadastro do novo curso.
6	MIT - 1A2	01	FP10	Não é apresentada uma mensagem informando se o cadastro do novo curso foi realizado com sucesso.
7	MIT - 1C1	01	E1.3	Quando o usuário cadastra ou altera um curso para um já existente o sistema não dá a possibilidade do usuário refazer sua ação.
8	MIT - 1E1	01	E1.2	Não é apresentado ao usuário o que define que um curso já existe.
9	MIT - 1H3	01	E2.1.1	O sistema não informa qual campo está vazio.
10	MIT - 1H3	01	E2.3.1	O sistema apresenta uma mensagem de forma que possa inibir ou desagradar o usuário.
11	MIT - 1H2	01	E2.3.1	O sistema apresenta de formas diferentes mensagens que dizem a mesma coisa, porém de campos diferentes.

Figura 11 – Planilha de discrepâncias de um inspetor durante o uso da MIT 1

Inspetor:	YYYYY		
Tempo de inspeção:	64 minutos		
Nro	Heurística	Nro do Mockup	Descrição do Problema Encontrado
1	MIT - 2E1	01	Não são claramente definidos quais são os campos obrigatórios.
2	MIT - 2E2	01	Não são indicados os formatos das datas de início e término.
3	MIT - 2G5	01	A escolha de um curso é feita de forma manual, o usuário precisa digitar o nome e pesquisar, isso se torna muito trabalhoso.
4	MIT - 2G4	01	A definição de status é feita de forma manual, o usuário precisa digitar, isso é muito trabalhoso.
5	MIT - 2G4	01	A definição da Unidade ou Centro de Formação é feita de forma manual, o usuário precisa digitar, isso é muito trabalhoso, o mesmo acontece para Instituições Parceiras.

Figura 12 – Planilha de discrepâncias de um inspetor durante o uso da MIT 2

Inspetor:	AAAAAAA		
Tempo de inspeção:	30 minutos		
Nro	Heurística	Nro do Diagrama de Atividade	Descrição do Problema Encontrado
1	MIT - 3C1	01	Botões "Próximo" e "Seguinte" parecem fazer a mesma coisa com nomes diferentes na tela "Busca de Curso".
2	MIT - 3A1	01	Na tela "Busca de Curso" não deixa claro ao funcionário porque o mesmo deve selecionar outras informações ou clicar em botões adicionais para ver o resultado da sua busca.
3	MIT - 3E1	01	A primeira atividade do diagrama não diz que opções devem ser exibidas, ou como devem ser exibidas.
4	MIT - 3H2	01	É mostrada a tela "Cadastro/Alteração" de curso quando o funcionário seleciona a opção Curso. Não é perguntado do funcionário se este realmente quer "Cadastrar/Alterar". E se ele quiser Listar ou Excluir?
5	MIT - 3B2	01	Não é possível excluir ou alterar cursos, na prática.

Figura 13 – Planilha de discrepâncias de um inspetor durante o uso da MIT 3

Após o estudo, as listas de discrepâncias individuais foram então integradas a uma única lista, retirando-se a referência do inspetor e da técnica utilizada por ele. A lista sem estas referências foi passada para outro pesquisador da área de IHC (autor do Caso de Uso e dos *Mockups*), que durante uma reunião decidiu quais dessas discrepâncias eram únicas e quais eram duplicatas (discrepâncias equivalentes apontadas por mais de um inspetor), além de classificar em defeitos ou falso-positivos.

#### 4.2.3 Análise Quantitativa

De acordo com o objetivo do estudo definido anteriormente, para verificar a viabilidade do conjunto de técnicas MIT, é necessário primeiramente verificar se ela alcança o seu objetivo de detectar defeitos. Para isso, mediu-se o número de defeitos encontrados por cada inspetor na avaliação de cada um dos modelos.

Além disso, para realizar esta avaliação dois indicadores quantitativos foram definidos: eficácia (razão entre o número de defeitos detectados e o total de defeitos existentes) e eficiência (razão entre o número de defeitos detectados e o tempo gasto na inspeção).

o Avaliação de Usabilidade em Caso de Uso

Ao analisar a Tabela 13 pode-se verificar que a MIT 1 ajudou os inspetores a detectarem mais defeitos na inspeção. Pode-se observar também que os inspetores que usaram a MIT 1 tiveram o tempo de inspeção variando entre 0,88 horas e 2,20 horas e encontraram entre 13 e 23 defeitos. Já os inspetores que usaram a AH empregaram entre 1,05 a 1,60 horas na detecção, encontrando entre 8 e 14 defeitos.

**Tabela 13 - Resultados por Inspetor – MIT 1 X AH**

Técnica/ Inspetor	Número de Discrepâncias	Número de Falso- Positivo	Número de Defeitos	Tempo em Horas	Defeitos por Hora	Total de Defeitos
MIT 1	1	21	3	18	1,67	10,80
	2	17	2	15	1,30	11,54
	3	19	4	15	1,72	8,74
	4	14	1	13	1,38	9,40
	5	19	4	15	0,88	16,98
	6	23	2	21	2,20	9,55
	7	18	2	16	1,33	12,00
	8	32	9	23	1,50	15,33
AH	9	15	2	13	1,43	9,07
	10	15	5	10	1,60	6,25
	11	18	6	12	1,35	8,89
	12	12	1	11	1,27	8,68
	13	18	4	14	1,05	13,33
	14	13	0	8	1,42	5,65
	15	22	9	13	1,28	10,13

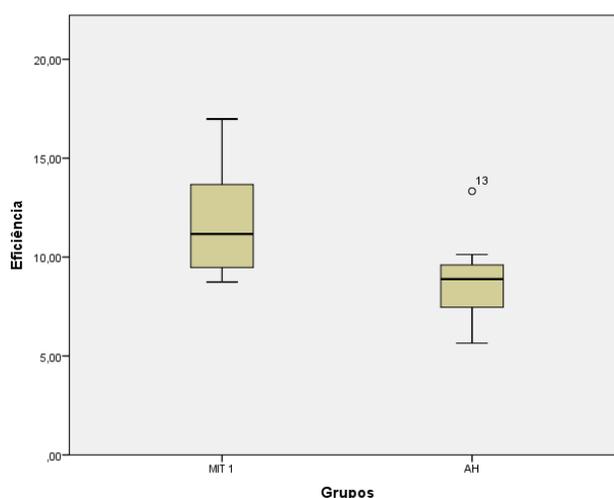
Nesta inspeção foi identificado um total de 103 defeitos de usabilidade sem duplicatas (número muito superior ao número de defeitos inseridos que foram 11). A Tabela 14 a seguir apresenta as médias calculadas para os indicadores de eficácia e eficiência.

**Tabela 14 - Resultados dos indicadores de Eficiência e Eficácia – MIT 1 X AH**

Técnica	Total de Defeitos	Média de Defeitos	Eficácia Média	Tempo Total em Horas	Eficiência Média (Defeitos por Hora)
MIT 1	136	17,00	17%	11,98	11,35
AH	81	11,57	11%	9,40	8,62

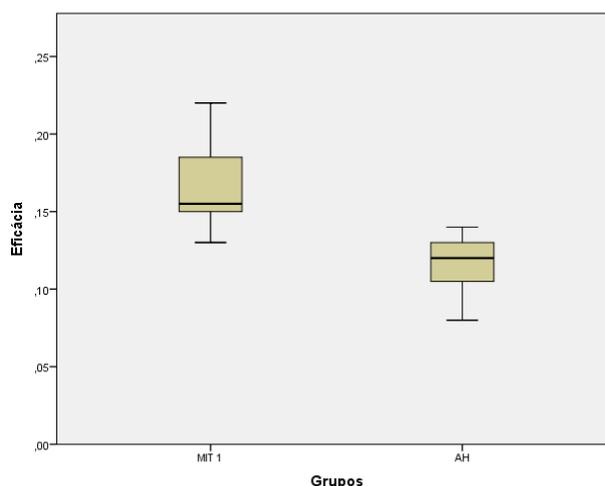
Para comparar a eficiência das duas amostras, utilizou-se análise de *boxplots* e o teste não-paramétrico Mann-Whitney (MANN e WHITNEY 1947). Para realização da análise estatística foi utilizado o *software* SPSS v 20.0 (SPSS 2011) com  $\alpha=0.10$ . A Figura 14 mostra os *boxplots* comparando a distribuição de eficiência por técnica.

Ao analisar a Figura 14, pode-se notar que a mediana do Grupo MIT 1 está mais alta que a mediana do Grupo AH. Além disso, ao comparar as duas amostras usando o teste Mann-Whitney, foi encontrada diferença significativa entre os dois grupos ( $p = 0.0541$ ). Esses resultados sugerem que o Grupo MIT 1 foi mais eficiente que o Grupo AH ao inspecionar a usabilidade no caso de uso.



**Figura 14** – Boxplots de eficiência por técnica - MIT 1 X AH

A mesma análise foi aplicada para verificar se houve diferença significativa em relação ao indicador eficácia das duas técnicas na detecção de defeitos de usabilidade. Os *boxplots* apresentados na Figura 15 mostram que o grupo de inspetores que utilizaram a MIT 1 obteve um desempenho significativamente superior que o do grupo que utilizou a AH. Isto foi confirmado pelo teste Mann-Whitney ( $p = 0,0012$ ). Estes resultados apontam que a técnica MIT 1 foi mais eficaz do que a AH para inspecionar o caso de uso Manutenção de Cursos.



**Figura 15** - Boxplots de eficácia por técnica - MIT 1 X AH

○ Avaliação de Usabilidade em *Mockups*

Observando a Tabela 15 a seguir, pode-se notar que a MIT 2 ajudou os inspetores a detectarem mais defeitos na inspeção do que a AH. Porém a diferença entre a quantidade de defeitos encontrados por estes dois grupos é não significativa estatisticamente.

**Tabela 15** - Resultados por Inspetor – MIT 2 X AH

Técnica/ Inspetor	Número de Discrepâncias	Número de Falso- Positivo	Número de Defeitos	Tempo em Horas	Defeitos por Hora	Total de Defeitos
MIT 2	1	16	5	11	1,07	10,31
	2	26	11	15	1,07	14,06
	3	8	1	7	0,77	9,13
	4	13	2	11	0,63	17,37
	5	9	1	8	0,42	19,20
	6	14	1	13	1,02	12,79
	7	15	9	6	0,53	11,25
	8	22	8	14	0,67	21,00
AH	9	10	2	8	0,57	14,12
	10	6	0	6	0,42	14,40
	11	13	6	7	0,63	11,05
	12	13	4	9	0,55	16,36
	13	21	4	17	1,15	14,78
	14	19	7	12	0,98	12,20
	15	11	4	7	0,48	14,48

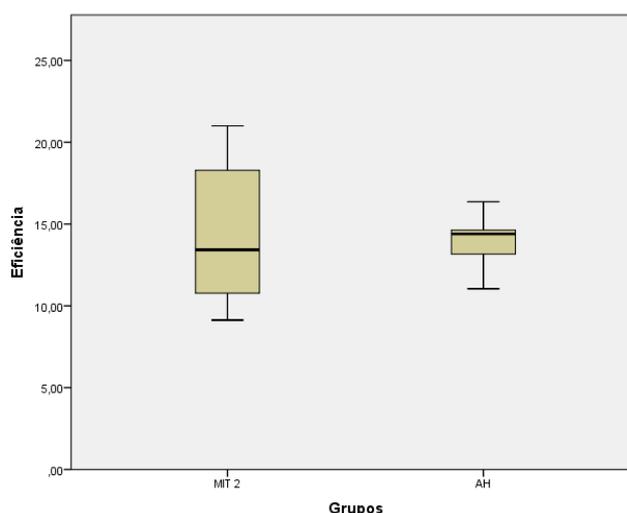
Um total de 66 defeitos de usabilidade sem duplicatas foi identificado nos *mockups* (sendo que apenas 6 defeitos foram inseridos). A Tabela 16 a seguir apresenta as médias para os indicadores de eficácia e eficiência.

**Tabela 16** - Resultados dos indicadores de Eficiência e Eficácia – MIT 2 X AH

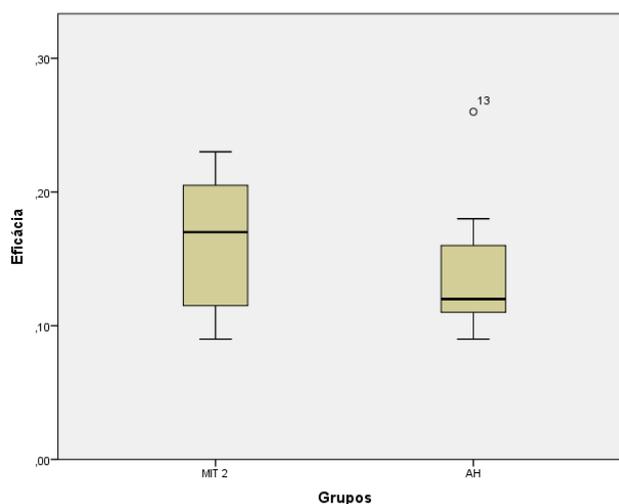
Técnica	Total de Defeitos	Média de Defeitos	Eficácia Média	Tempo Total em Horas	Eficiência Média (Defeitos por Hora)
MIT 2	85	10,63	16%	6,17	13,78
AH	66	9,43	14%	4,78	13,80

Para comparar a eficiência das duas amostras, utilizou-se novamente a análise de *boxplots* e o teste não-paramétrico Mann-Whitney. Para realização da análise estatística também foi utilizado o *software* SPSS com  $\alpha=0.10$ . A Figura 16 mostra os *boxplots* comparando a distribuição de eficiência por técnica.

Na Figura 16, pode-se notar que a mediana do Grupo AH está um pouco mais alta que a mediana do Grupo MIT 2. Ao comparar as duas amostras usando o teste Mann-Whitney, não foi encontrada diferença significativa entre os dois grupos ( $p = 0.8665$ ). Esses resultados sugerem que MIT 2 e AH proveram eficiência similar quando utilizados para inspecionar os *mockups* Cadastro de Turmas e Listagens de Turmas.

**Figura 16** - Boxplots de eficiência por técnica - MIT 2 X AH

Uma análise foi aplicada para verificar se houve diferença significativa em relação ao indicador eficácia das duas técnicas na detecção de defeitos de usabilidade em *mockups*. Através da mediana dos *boxplots* apresentados na Figura 17 pode-se notar que a mediana do Grupo MIT 2 está um pouco mais alta que a mediana do Grupo AH. Isso mostra que o grupo de inspetores que utilizaram a MIT 2 obteve um desempenho um pouco maior que o do grupo que utilizou a AH. Porém, ao comparar as duas amostras usando o teste Mann-Whitney, não foi encontrada diferença significativa entre os dois grupos ( $p = 0.5358$ ). Esses resultados sugerem que MIT 2 e AH proveram eficácia similar quando utilizados para inspecionar os *mockups* Cadastro de Turmas e Listagens de Turmas.



**Figura 17** - Boxplots de eficácia por técnica - MIT 2 X AH

○ Avaliação de Usabilidade em Diagrama de Atividades

Pode-se verificar através da Tabela 17 que a MIT 3 ajudou os inspetores a detectarem mais defeitos do que a AH.

**Tabela 17** - Resultados por Inspetor – MIT 3 X AH

Técnica/ Inspetor	Número de Discrepâncias	Número de Falso- Positivo	Número de Defeitos	Tempo em Horas	Defeitos por Hora	Total de Defeitos
<b>MIT 3</b>	1	10	1	9	0,68	13,17
	2	8	1	7	0,72	9,77
	3	9	0	9	0,57	15,88
	4	8	0	8	0,83	9,60
	5	10	3	7	0,50	14,00
	6	11	2	9	0,50	18,00
	7	7	3	4	0,30	13,33
	8	12	3	9	0,93	9,64
<b>AH</b>	9	8	3	5	1,08	4,62
	10	9	4	5	0,68	7,32
	11	2	0	2	0,30	6,67
	12	17	4	13	1,00	13,00
	13	6	1	5	1,00	5,00
	14	6	1	5	0,58	8,57
	15	9	4	5	0,58	8,57

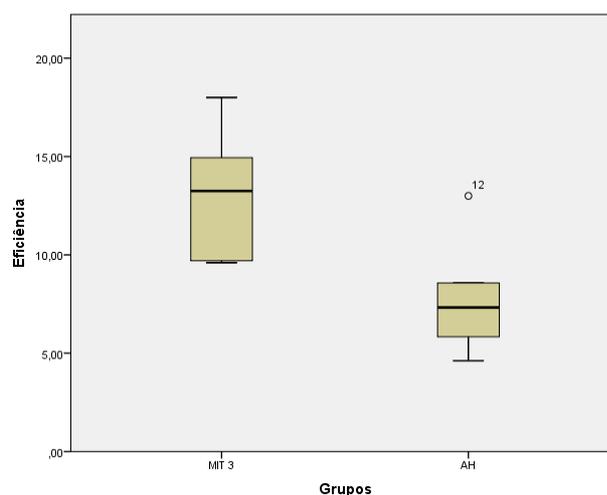
Nesta inspeção foi identificado um total de 27 defeitos de usabilidade sem duplicatas no diagrama de atividades (sendo que apenas 6 defeitos foram inseridos). A Tabela 18 a seguir apresenta as médias para os indicadores eficácia e eficiência. Pode-se notar que a média de tempo gasto pelos inspetores que utilizaram a MIT 3 está relativamente próxima da média de tempo gasto pelos inspetores da AH.

**Tabela 18** - Resultados dos indicadores de Eficiência e Eficácia – MIT 3 X AH

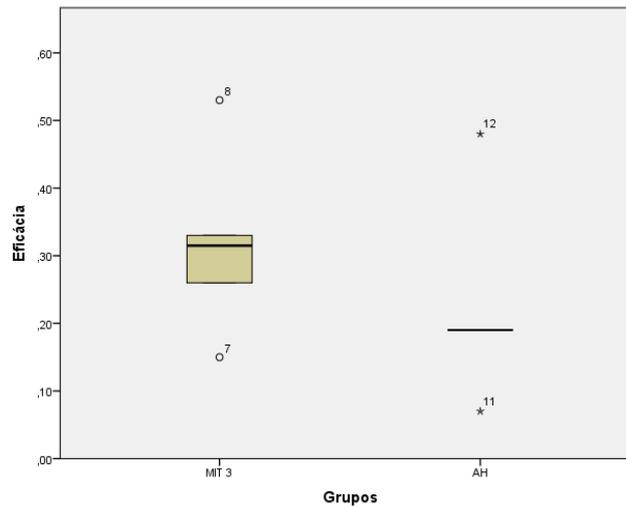
Técnica	Total de Defeitos	Média de Defeitos	Eficácia Média	Tempo Total em Horas	Eficiência Média (Defeitos por Hora)
MIT 3	62	7,75	29%	5,03	12,32
AH	40	5,71	21%	5,23	7,64

Para comparar a eficiência das duas amostras, também utilizou-se *boxplots* e o teste não-paramétrico Mann-Whitney. Para realização da análise estatística também foi utilizado o *software* SPSS com  $\alpha=0.10$ . A Figura 18 mostra os *boxplots* comparando a distribuição de eficiência por técnica.

Pode-se notar que a mediana do Grupo MIT 3 está mais alta que a mediana do Grupo AH. E ao comparar as duas amostras usando o teste Mann-Whitney, foram encontradas diferenças significativas entre os dois grupos ( $p = 0.0022$ ). Esses resultados sugerem que a MIT 3 é mais eficiente que a AH para detecção de defeitos em diagramas de atividades.

**Figura 18** - Boxplots de eficiência por técnica - MIT 3 X AH

Foi aplicada também uma análise para verificar se houve diferença significativa em relação ao indicador eficácia das duas técnicas na detecção de defeitos de usabilidade. Através da mediana dos *boxplots* apresentados na Figura 19 pode-se notar que a mediana do Grupo MIT 3 está mais alta que a mediana do Grupo AH. Isso mostra que o grupo de inspetores que utilizaram a MIT 3 obteve um desempenho um pouco maior que o do grupo que utilizou a AH. Ao comparar as duas amostras usando o teste Mann-Whitney, foi encontrada diferença significativa entre os dois grupos ( $p = 0.0721$ ). Esses resultados sugerem que MIT 3 possui eficácia maior que AH quando utilizados para inspecionar o diagrama de atividades Manutenção de Cursos.



**Figura 19** - Boxplots de eficácia por técnica - MIT 3 X AH

Neste primeiro estudo experimental realizado, os resultados quantitativos obtidos mostram que: a MIT 1 apresentou uma eficiência de 11,35 defeitos por hora, e eficácia de 17%, já a AH mostrou uma eficiência de 8,62 defeitos por hora e eficácia de 11%. De acordo com esses resultados, é possível ter indícios de que o uso da MIT 1 como técnica de inspeção de usabilidade em casos de usos é viável para a detecção de defeitos.

A MIT 2, mostrou uma eficiência de 13,78 defeitos por hora, e eficácia de 16%, enquanto a AH eficiência de 13,80 defeitos por hora e eficácia de 14%. Embora estes resultados demonstrem que a MIT 2 apoia a detecção de defeitos, eles também apontam que, para a versão atual, não há diferença entre o emprego da MIT e da AH. Este pode ser um indicativo de que em *mockups* podem ser usadas técnicas de inspeção tradicionais, como a AH.

E por fim a MIT 3 apresentou uma eficiência de 12,32 defeitos por hora, e eficácia de 29% e a AH mostrou eficiência de 7,64 defeitos por hora e eficácia de 21%. De acordo com esses resultados, é possível ter indícios de que o uso da MIT 3 como técnica de inspeção de usabilidade de diagrama de atividades é viável para a detecção de defeitos.

No entanto, devido à pequena amostra, não é possível considerar estes resultados conclusivos, sendo necessário repetir este estudo com uma amostra maior e mais heterogênea de participantes.

#### 4.2.4 Análise Qualitativa

Para MALLARDO e CALEFATO (2003) investigar a aceitação dos usuários para uma tecnologia requer um modelo que explique as atitudes e comportamentos das pessoas. Neste estudo utilizou-se o modelo de aceitação de tecnologia (*Technology Acceptance Model - TAM*) proposto por DAVIS (1989) para a realização da análise qualitativa. A estrutura do

modelo TAM tem como base dois fatores: (1) Percepção sobre utilidade, definida como “o grau no qual uma pessoa acredita que utilizar uma tecnologia específica melhoraria seu desempenho no trabalho” e (2) Percepção sobre facilidade de uso, definida como “o grau no qual uma pessoa acredita que utilizar uma tecnologia específica seria livre de esforço” (LAITENBERGER e DREYER 1998).

Foram utilizados na análise qualitativa os questionários preenchidos após a inspeção, onde os participantes relataram suas ações e impressões em relação à utilização de cada MIT. A Tabela 19 apresenta as questões e afirmações do questionário pós-inspeção e o Apêndice A apresenta este questionário completo.

**Tabela 19** - Questões e afirmações do questionário pós-inspeção

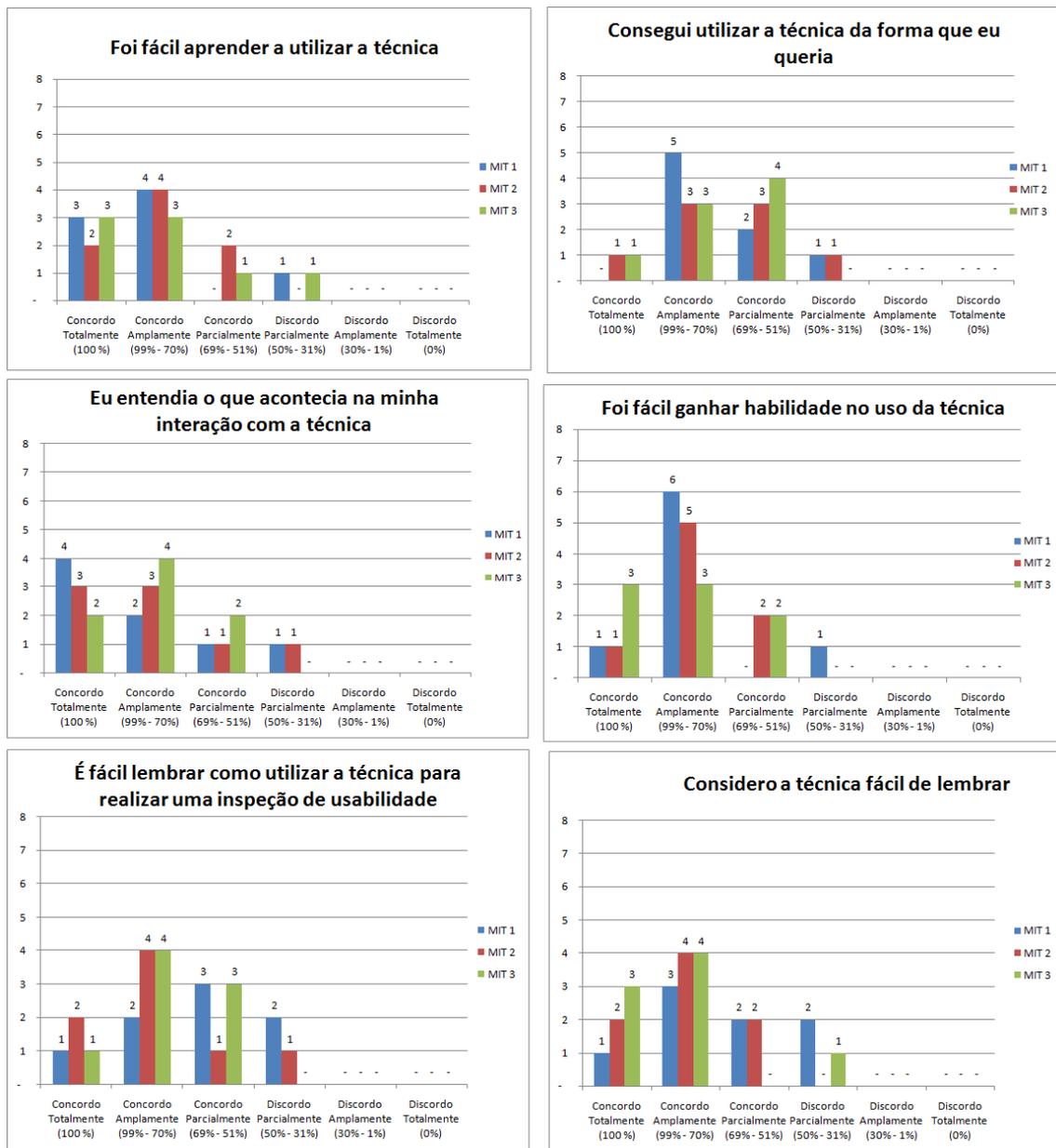
Questões	Afirmações
1. Em relação à sua percepção sobre a facilidade de uso da técnica MIT, qual o seu grau de concordância em relação às seguintes afirmações?	1. Foi fácil aprender a utilizar a MIT; 2. Consegui utilizar a MIT da forma que eu queria; 3. Eu entendia o que acontecia na minha interação com a MIT; 4. Foi fácil ganhar habilidade no uso da MIT; 5. É fácil lembrar como utilizar a MIT para realizar uma inspeção de usabilidade; 6. Considero a MIT fácil de usar.
2. Em relação à sua percepção sobre a utilidade da técnica MIT, qual o seu grau de concordância em relação às seguintes afirmações?	1. A MIT me permitiu detectar defeitos mais rápido; 2. Usar a MIT melhorou o meu desempenho na inspeção (acredito ter encontrado um número maior de defeitos do que encontraria sem utilizar a MIT); 3. Usar a MIT facilitou a inspeção; 4. Eu considero a MIT útil para inspeções de usabilidade.
3. Em relação a uma questão específica da técnica MIT, qual o seu grau de concordância em relação à seguinte afirmação?	1. As heurísticas da técnica MIT são fáceis de compreender

Uma escala de seis pontos foi utilizada, tendo como base os questionários aplicados por MALLARDO e CALEFATO (2003), que são: concordo totalmente, concordo amplamente, concordo parcialmente, discordo parcialmente, discordo amplamente e discordo totalmente. Nesse questionário, os inspetores respondiam qual o seu grau de concordância em relação à utilidade e facilidade de uso, além do seu grau de concordância em relação à facilidade de compreender.

o Percepção sobre Facilidade de Uso

A Figura 20 exhibe as respostas referentes à Percepção sobre facilidade de uso da MIT 1, 2 e 3. Pode-se destacar que na MIT 1 dois dos oito inspetores discordaram das questões “*é fácil lembrar como utilizar a MIT 1 para realizar uma inspeção de usabilidade*” e

“considero a MIT 1 fácil de usar”, mostrando que houveram dificuldades ao usar e lembrar da MIT 1.

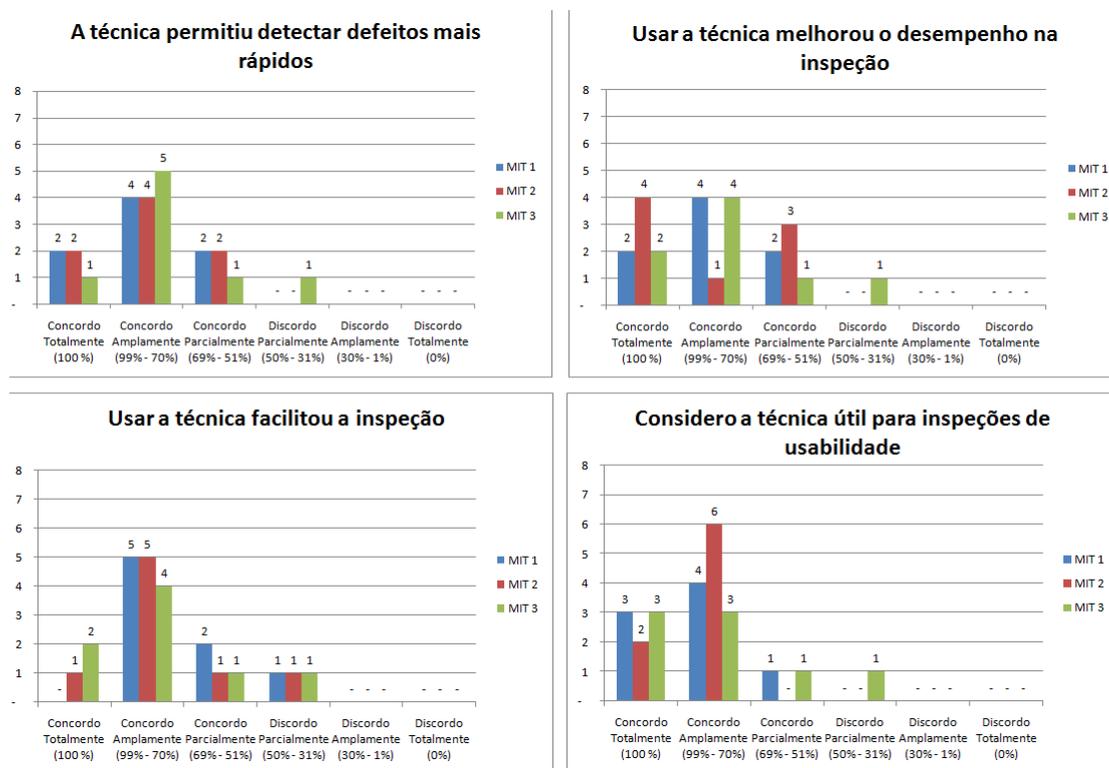


**Figura 20** - Resultado das Questões relacionadas à Percepção sobre a facilidade de uso das MITs

o Percepção sobre Utilidade da Técnica

As respostas relacionadas à Percepção sobre utilidade da MIT 1, 2 e 3 são exibidas na Figura 21. As questões relatam sobre a técnica permitir detectar defeitos mais rápido, se usar ajuda a melhorar o desempenho na inspeção, se usar facilita a inspeção e se considera útil para inspeções de usabilidade. Observa-se que um usuário discordou parcialmente em todos os questionamentos da MIT 3, o que indica que em algum momento o usuário encontrou dificuldade de utilização. Outro ponto que pode ser destacado é que tanto na MIT 1 quanto na MIT 2 um inspetor discordou parcialmente do questionamento “usar a

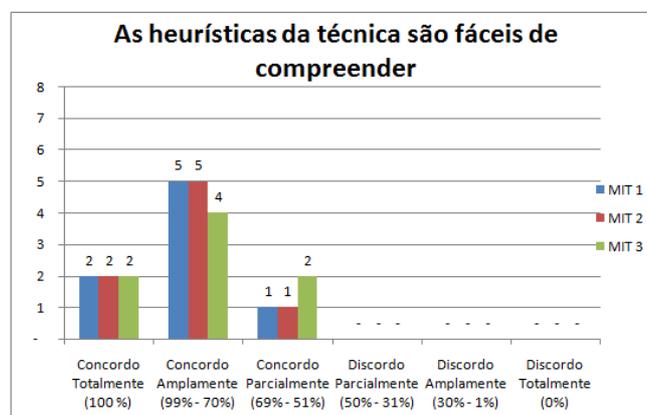
*técnica facilitou a inspeção*”, indicando que utilizar estas duas técnicas não auxilia o processo de inspeção.



**Figura 21** - Resultado das Questões relacionadas à Percepção sobre a utilidade das MITs.

o Percepção sobre Facilidade de Compreender

Através do questionário pós-inspeção foi adicionado um fator que não é apresentado no modelo TAM, chamado de Facilidade de Compreensão, que contém uma pergunta específica sobre a compreensão da técnica. As respostas relacionadas à Facilidade de Compreensão da MIT 1, 2 e 3 são exibidas na Figura 22, onde se destaca a boa avaliação dada às três técnicas pelos inspetores. Nota-se que não houve nenhum inspetor que discordasse da questão, mostrando que no geral as três técnicas são fáceis de compreender.



**Figura 22** - Resultado das Questões relacionadas à Facilidade de Compreensão das MITs

Além da análise através do modelo TAM, foi feita uma análise específica dos dados qualitativos que foram coletados por meio dos questionários pós-inspeção, tendo como base o método *Grounded Theory* (GT) (STRAUSS e CORBIN 1998). Analisaram-se os dados qualitativos utilizando um subconjunto das fases do processo de codificação sugerida por STRAUSS e CORBIN (1998) – as codificações aberta e axial. Como o objetivo não era criar uma teoria acerca da percepção dos inspetores sobre a experiência de usar as MITs, não foi realizada a etapa 3 do método GT (codificação seletiva).

No processo de codificação são identificados códigos e categorias. Um código dá nome a um fenômeno de interesse para o pesquisador; abstrai um evento, objeto, ação, ou interação que tem um significado para o pesquisador (STRAUSS e CORBIN 1998). Categorias são agrupamentos de conceitos unidos em um grau de abstração mais alto.

Os códigos são apresentados seguidos de dois números que representam respectivamente o grau de fundamentação (*groundedness*) e o de densidade teórica (*density*) do código. O grau de fundamentação (*groundedness*) mostra o número de citações com as quais o código está associado. O grau de densidade teórica (*density*) mostra o número de relacionamentos do código com outros códigos.

Os códigos desta análise foram criados com base em citações dos inspetores presentes nos questionários. No total foram identificados 23 códigos na MIT 1, 16 códigos na MIT 2 e 19 códigos na MIT 3. Em cada técnica os códigos foram agrupados em duas categorias: (i) estrutura da técnica, e (ii) facilidade da aplicação.

#### o Categoria Estrutura da Técnica

A categoria "Estrutura da técnica" apresenta os códigos derivados dos comentários dos inspetores a respeito da organização dos elementos da técnica.

Na MIT 1 sete códigos estão associados a esta categoria, ou seja o grau de fundamentação é 7 (Figura 23). Fazendo uma análise a partir do esquema gráfico, observam-se na Figura 23 evidências de dificuldade a partir dos códigos “*Em algumas partes há ambiguidades*” e “*Algumas heurísticas são muito parecidas*”. Dois diferentes inspetores relataram que “*Algumas heurísticas causam confusão/indecisão*”, indicando que algumas heurísticas necessitam ser melhoradas. Além disso, existem evidências de inadequação como o código “*Não foi encontrada utilização da MIT – 1E*”, e como houve sugestões para melhorar a técnica criou-se uma subcategoria chamada “*Sugestões de Melhoria*” onde fazem parte os códigos “*Sugestão de generalizar algumas heurísticas para tornar a inspeção mais fácil*”, “*Sugestão de criação de itens para tratar ambiguidades*” e “*Sugestão de Criação de itens para avaliar se dados obrigatórios e opções disponíveis estão claros*”.

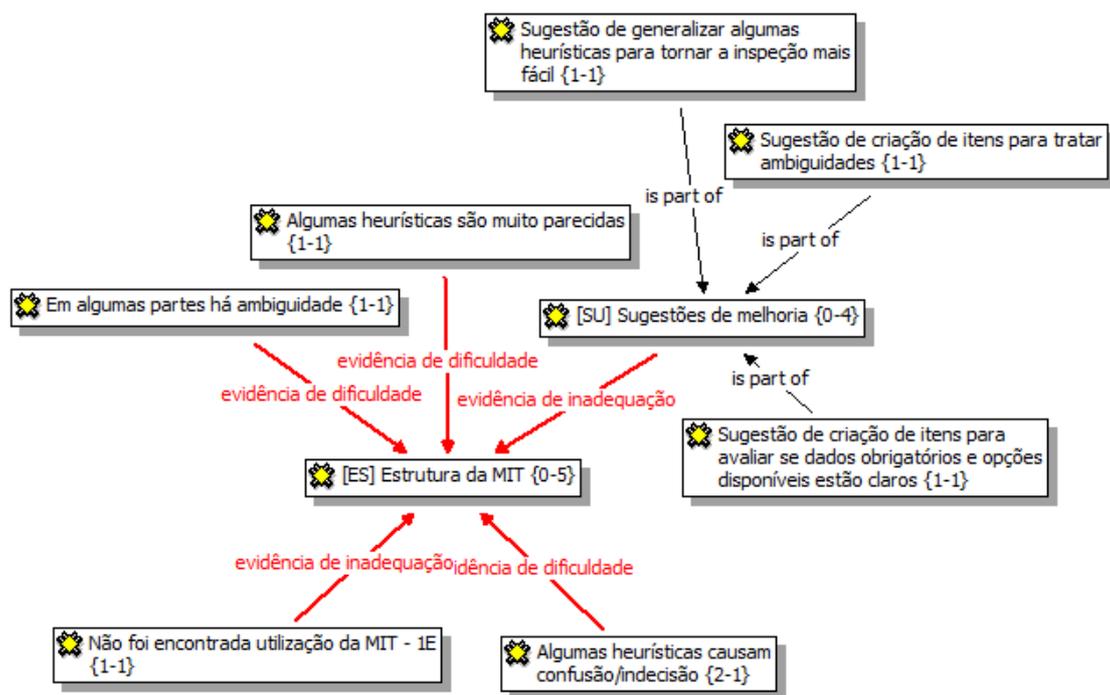


Figura 23 - Esquema gráfico da categoria Estrutura da MIT 1

Na MIT 2 também sete códigos estão relacionados a categoria Estrutura da MIT 2 (Figura 24). A partir do esquema gráfico, pode-se notar que existem evidências de inadequação como os códigos “Necessidade de especificar melhor o item MIT – 2F4”, “MIT 2 com heurísticas amplas mas com poucas ocorrências” e “Há itens desnecessários por se tratar de uma avaliação em mockup impresso”. Além disso, na subcategoria “Sugestões de Melhoria” houve os seguintes códigos: “Sugestão de uma preparação adequada dos itens”, “Sugestão de compreender quando cada MIT apresenta características singulares e quando for feita a inspeção em único modelo, esta MIT possuir todas as características”, “Sugestão de generalizar mais as heurísticas” e “Sugestão de adaptar ao mockup os itens de avaliação que no momento são desnecessários”, sendo que neste último código, observa-se através do grau de fundamentação igual a dois que dois diferentes inspetores sugeriram isso.

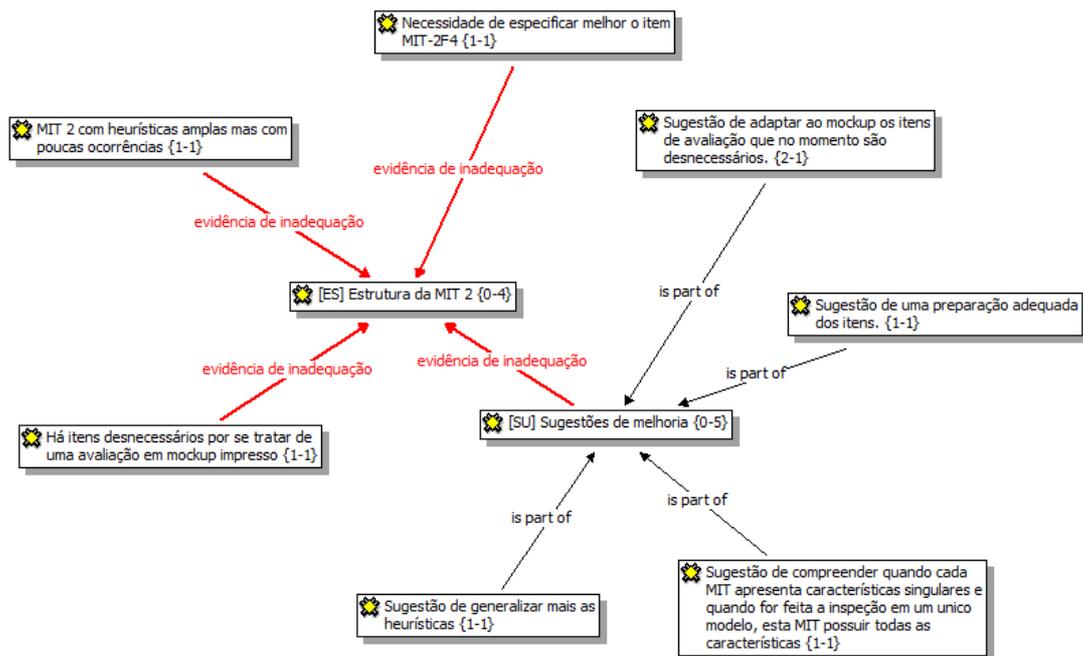


Figura 24 - Esquema gráfico da categoria Estrutura da MIT 2

E por fim na MIT 3 seis códigos estão associados a categoria Estrutura da MIT 3 (Figura 25). Através do esquema gráfico, observa-se que existem evidências de dificuldade como o código “MIT 3 possui heurísticas parecidas” e “Sugestão de criar heurísticas menos genéricas e mais definidas”. Além disso, houve evidências de inadequação como os códigos “Alguns erros da MIT 3G podem ser diferentes dos citados” (onde o inspetor quis dizer que além dos exemplos citados nos itens de verificação da MIT havia outros), “Necessidade de desmembrar mais a MIT 3G” e o código “Necessidade de criar uma opção na MIT 3B de cancelar, voltar ou desfazer uma atividade” e o código “Necessidade de criar uma opção na MIT 3B de cancelar uma atividade”.

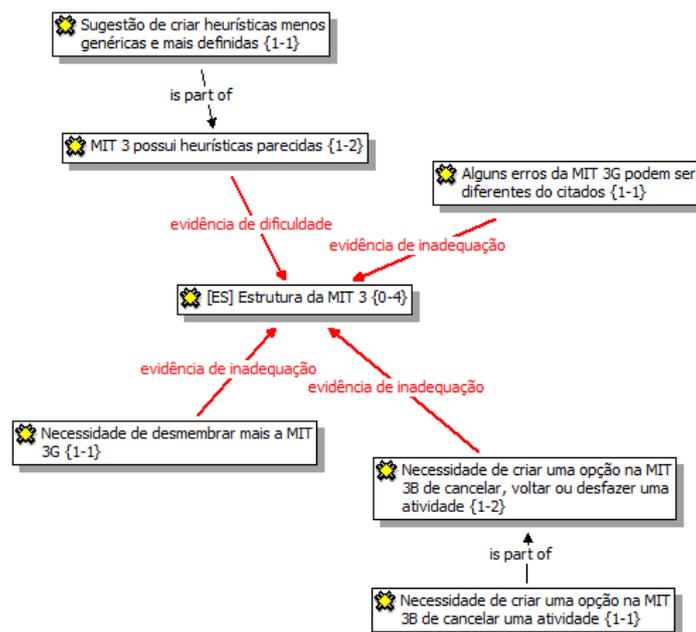


Figura 25 - Esquema gráfico da categoria Estrutura da MIT 3

o Categoria Facilidade da Aplicação

A categoria "Facilidade da aplicação" apresenta os códigos derivados dos comentários dos inspetores sobre aspectos que facilitam ou dificultam a aplicação da técnica. Na MIT 1 dezesseis códigos estão associados a esta categoria (Figura 26).

Os códigos associados a evidências de dificuldades são “É difícil relacionar problemas a uma heurística”, “Não é fácil identificar qual a heurística correspondente”, “É difícil ser imparcial quanto à utilização das heurísticas”, “Não é fácil encontrar as heurísticas por causa da estrutura do documento de impressão”, “MIT 1 é fácil de lembrar com consulta” e “MIT 1 é fácil de lembrar”.

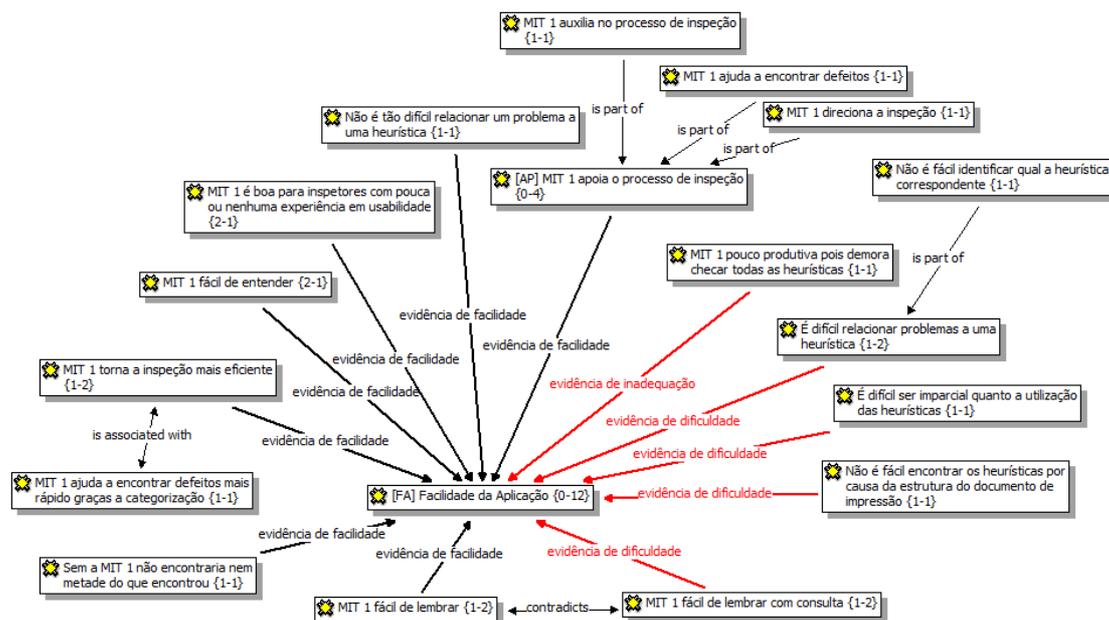


Figura 26 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 1

Já a MIT 2 possui nove códigos associados a categoria Facilidade da Aplicação (Figura 27). Através do esquema gráfico, notam-se evidências de inadequação a partir dos códigos “Grande quantidade de heurísticas torna a inspeção demorada, menos produtiva e chata” e “MIT 2 causa fadiga e cansaço”. Além disso, há uma evidência de dificuldade observada através do código “É difícil ser imparcial quanto à utilização das heurísticas”.

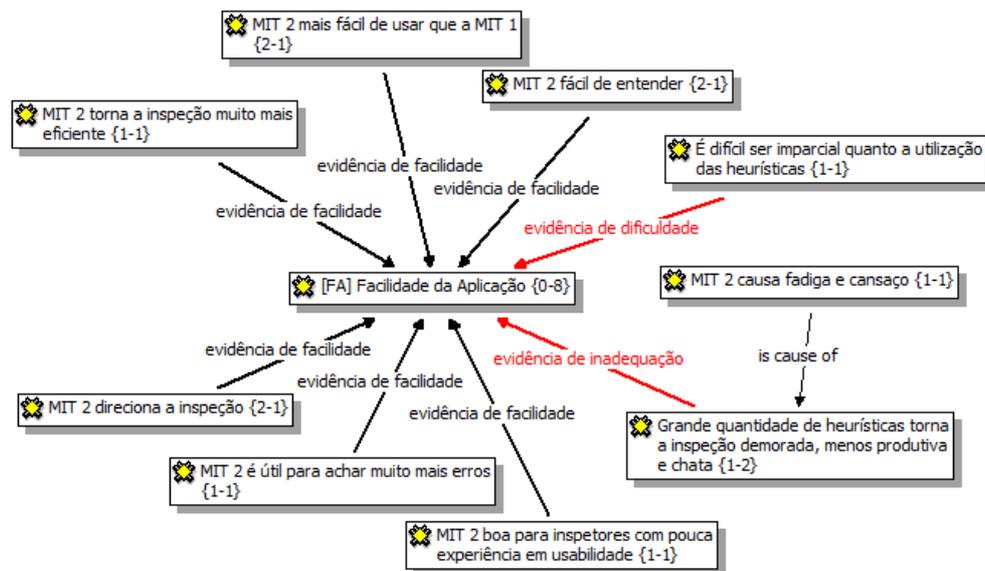


Figura 27 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 2

E por fim na MIT 3 há treze códigos associados a categoria Facilidade da Aplicação (Figura 28). A partir do esquema gráfico, observam-se evidências de dificuldade a partir dos códigos “É difícil ser imparcial quanto a escolha da heurística na hora da classificação por causa da tendência de usar determinada heurística para maioria dos casos” e “MIT 3 não é tão fácil de lembrar. Além disso, há evidências de inadequação observada através dos códigos “Alguns erros ainda ficam escondidos”, “MIT 3 não ajudou a identificar nenhum novo problema, apenas confirmou o que o inspetor já sabia ser um problema” e “MIT 3 não contribuiu tanto”.

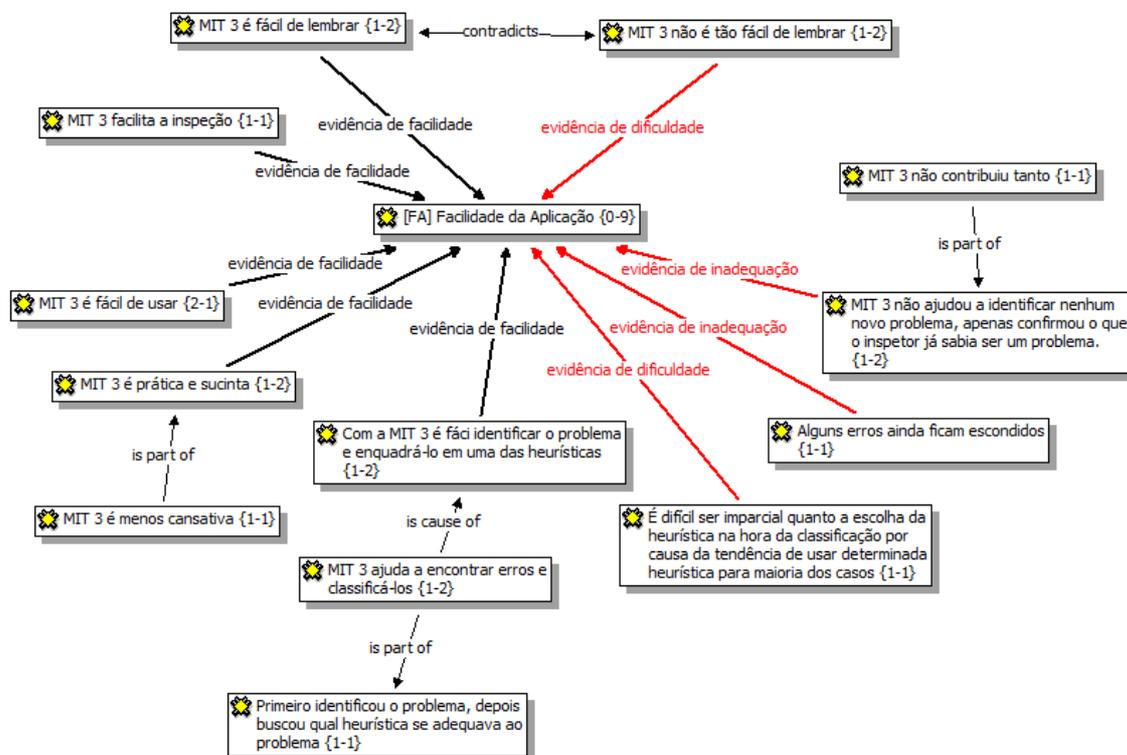
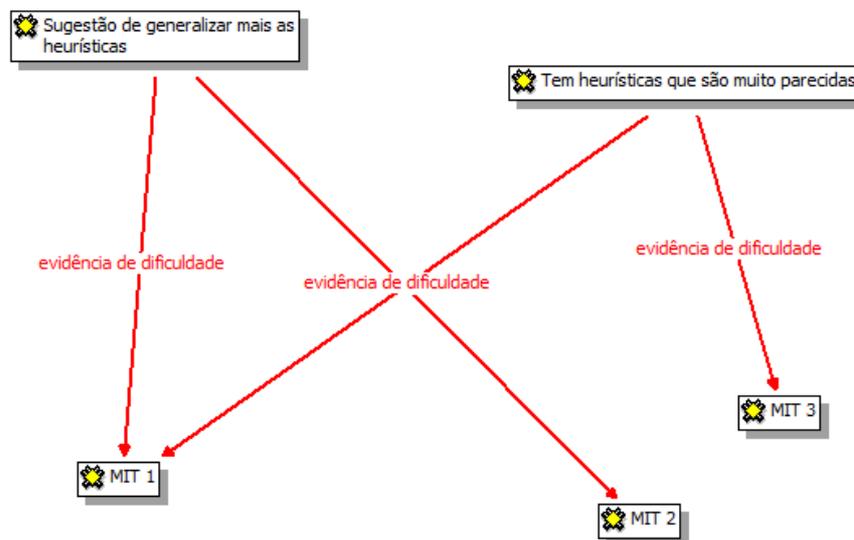


Figura 28 - Esquema gráfico da categoria Facilidade da Aplicação da MIT 3

- Análise de comparação

Com o intuito de fazer uma comparação dos códigos que se repetem em cada uma das técnicas realizou-se uma Análise de Comparação para cada uma das categorias. Através da análise desta comparação é possível verificar quais problemas são recorrentes e quais facilidades se repetem.

Através da Figura 29, nota-se que algumas evidências de dificuldade se repetem na categoria “Estrutura das Técnicas”. O código “*Sugestão de generalizar mais as heurísticas*” está presente na MIT 1 e MIT 2. Já o código “*Tem heurísticas que são muito parecidas*” aparece tanto na MIT 1 quanto na MIT 3.



**Figura 29** - Esquema gráfico da análise de comparação da categoria Estrutura das Técnicas

Já na Figura 30, podem-se observar evidências de dificuldade e facilidade que se repetem na categoria “Facilidade da Aplicação”. O código referente a evidência de dificuldade que se repete na MIT 1 e MIT 2 é “*Pouco produtiva pois demora checar todas as heurísticas*”. Já o código que se repete na MIT 1, MIT 2 e MIT 3 é “*É difícil ser imparcial quanto a escolha da heurística na hora da classificação por causa da tendência de usar determinada heurística para maioria dos casos*”. E por fim, o código “*Não é tão fácil lembrar*” se repete na MIT 1 e MIT 3.

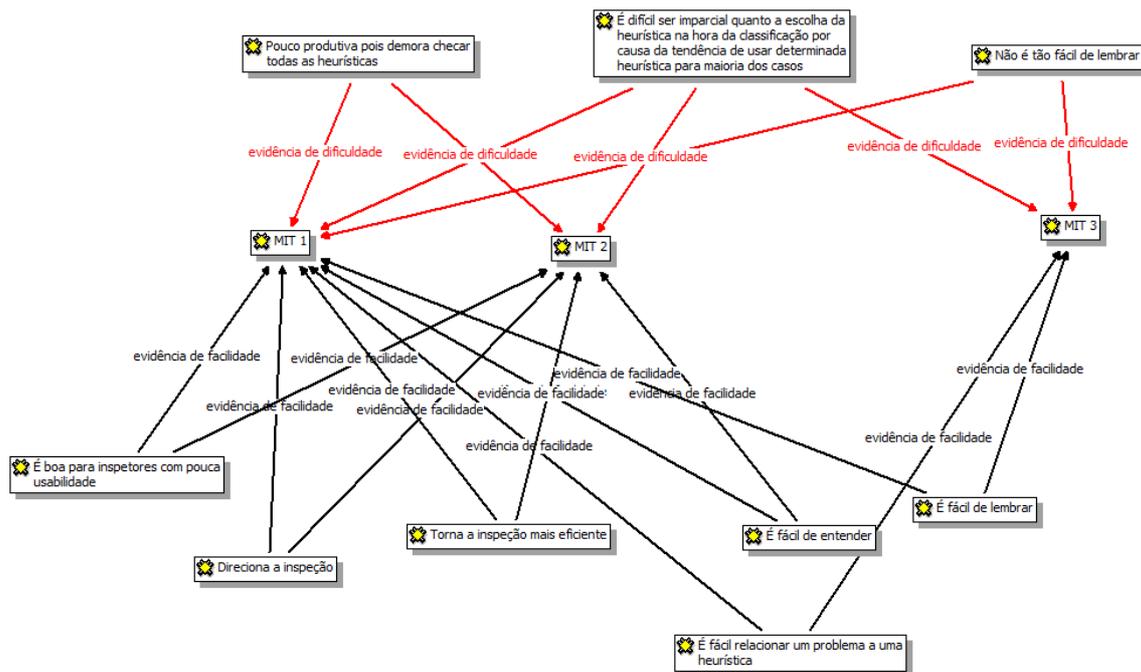


Figura 30 - Esquema gráfico da análise de comparação da categoria Facilidade da Aplicação

#### 4.2.5 Ameaças à Validade

Em todos os estudos experimentais existem ameaças que podem afetar a validade dos resultados. As ameaças relacionadas a este estudo são apresentadas a seguir, classificadas em quatro categorias: validade interna, validade externa, validade de conclusão e validade de constructo (WÖHLIN *et al.* 2000).

**Validade Interna:** neste estudo, consideraram-se três principais ameaças que representavam um risco de interpretação imprópria dos resultados: (1) efeitos de treinamento, (2) classificação de experiência e (3) medição de tempo. Em relação à primeira ameaça, poderia haver um efeito causado pelo treinamento, caso o treinamento da AH tivesse qualidade inferior ao treinamento das MITs. Este risco foi controlado preparando o mesmo treinamento sobre usabilidade para os dois grupos. Para o grupo das MITs foi realizado um experimento a parte contendo apenas instruções e exemplos das heurísticas das MITs. Em relação à classificação de experiência dos participantes, ela foi uma autoclassificação, com base em número e tipo de experiências anteriores (em desenvolvimento e conhecimento em usabilidade). Finalmente, sobre a medição do tempo, o moderador conferia o tempo anotado por cada participante a cada entrega da planilha com a descrição das discrepâncias.

**Validade Externa:** quatro questões foram consideradas: (1) os participantes do estudo foram estudantes de graduação; (2) o estudo foi realizado em ambiente acadêmico; (3) validade dos modelos como representante e (4) o própria autora da técnica inseriu os

defeitos nos modelos. Sobre a questão (1), alguns participantes possuíam experiência em aplicações na indústria e alguns não. Segundo CARVER *et al.* (2003), estudantes que não possuem experiência em aplicações na indústria podem apresentar habilidades similares a inspetores menos experientes, sendo este o perfil desejado para os participantes. Em relação à questão (2) os objetos da inspeção (Caso de Uso, *Mockup* e Diagrama de Atividades) são modelos que fazem parte da especificação de um sistema real. Sobre a questão (3), não é possível afirmar que os modelos utilizados na inspeção representem todo tipo de Casos de uso, *Mockup* e Diagramas de atividades. Em relação a questão (4), todos defeitos de usabilidade inseridos foram encontrados por participantes dos dois grupos. Além disso, o número de defeitos encontrados pelos inspetores tanto da MIT quanto da AH foi muito maior que a quantidade de defeitos inseridos pelo pesquisador.

**Validade de Conclusão:** neste estudo, o maior problema é o tamanho e a homogeneidade da amostra. Pois a quantidade de participantes não é o ideal do ponto de vista estatístico e por serem todos alunos de graduação de uma mesma instituição. Amostras reduzidas é um problema conhecido em estudos em IHC e ES (BONIFÁCIO *et al.* 2011, CONTE *et al.* 2007). Devido a estes fatos, há limitação nos resultados, sendo estes considerados indícios e não conclusivos.

**Validade de Constructo:** neste tipo de ameaça, considerou-se a definição dos indicadores eficiência e eficácia. Esses indicadores são comumente adotados em estudos que investigam técnicas de detecção de defeitos e estes indicadores foram medidos utilizando a mesma abordagem aplicada em BONIFÁCIO *et al.* 2011 e CONTE *et al.* 2009.

### 4.3 Considerações Finais

Este capítulo apresentou um estudo *in vitro* realizado para avaliar a viabilidade do conjunto de técnicas MIT. Neste estudo, os resultados apontam que a MIT 1 e MIT 3 ajudou os inspetores a encontrarem um número maior de defeitos de usabilidade, quando utilizada para inspecionar os modelos de um sistema real para Gerenciamento dos Cursos e Turmas de um Centro de Treinamento em comparação com a AH.

O próximo capítulo descreve a primeira evolução da técnica, baseada no resultado qualitativo e quantitativo do primeiro estudo de viabilidade.

## CAPÍTULO 5 - CONSTRUÇÃO DA VERSÃO ATUAL DO CONJUNTO DE TÉCNICAS MIT

### 5.1 Introdução

A proposta do conjunto de técnicas MIT é direcionar o processo de inspeção de usabilidade nos modelos de Caso de Uso, *Mockups* e Diagrama de Atividades. Entretanto, os dados qualitativos mostraram alguns problemas com a utilização das MITs, sobretudo em relação ao tempo gasto na fase de detecção, onde se verificou que as técnicas estavam causando desconforto e interferindo na produtividade, principalmente a MIT 2. A análise dos dados qualitativos permitiu a identificação dos problemas que geraram alterações na v1 das MITs. O texto completo da v2 das MITs (MIT v2) é apresentado no Apêndice B.

### 5.2 Melhorias na MIT 1 v1

Um inspetor notou através do seguinte código “*Sugestão de criação de itens para avaliar se dados obrigatórios e opções disponíveis estão claros?*” que não havia itens de verificação na MIT 1 que verificasse se as opções e os dados obrigatórios estavam disponível. Para resolver este problema, criou-se o seguinte item de verificação (Figura 31) que pertence à heurística 1AE – Prevenção de erros:

<b>1AE3</b>	Verifique se os dados ou opções obrigatórios FP, FA, FE e RN estão claramente definidos;
-------------	--

**Figura 31** - Item de verificação 1AE3 da Versão 2 da MIT 1

Outro inspetor observou que não haviam itens de verificação para analisar se há ambiguidades através do código “*Sugestão de criação de itens para tratar ambiguidades?*”. Portanto, uma nova item de verificação (Figura 32) foi acrescentada na MIT 1 relacionada à heurística “1AD - Consistência e padrões”:

<b>1AD3</b>	Verifique se os nomes de campos, telas, botões e links no FP, FA, FE e RN apresentam um único sentido, sem ambiguidades.
-------------	--

**Figura 32** - Item de verificação 1AD3 da Versão 2 da MIT 1

Além disso, um inspetor sugeriu o seguinte: “*Sugestão de generalizar algumas heurísticas para tornar a inspeção mais fácil?*”. Porém, se generalizássemos menos problemas poderiam ser encontrados através da MIT 1. Portanto, a solução utilizada para tornar a inspeção mais fácil foi tornar alguns itens de verificação que estavam extensos mais objetivos. Além disso,

esta modificação tratou de certa forma outras reclamações como: “*Algumas heurísticas são muito parecidas*”, “*Em algumas partes há ambiguidades*” e “*Algumas heurísticas causam indecisão/confusão*”.

Através do código “*Não foi encontrada utilização da MIT – 1E*” pôde-se notar que havia heurísticas que possuíam itens de verificação que não se adequavam ao tipo de caso de uso utilizado. Isto acontece porque há casos de uso com nível de detalhamento alto (possuem mensagens de erro, textos informativos, advertências, nome de telas, nome de campos, nome de botões, nome de links, nome de opções, entre outros). Além disso, há casos de uso com nível de detalhamento baixo, ou seja, não descreve as mensagens de erro, textos informativos, advertências, nome de telas, entre outros. Por este motivo, na versão 2 decidiu-se dividir a MIT 1 em Alto Detalhamento e Baixo Detalhamento, onde conforme a Figura 33 demonstra a MIT 1 - Alto Detalhamento é para casos de uso com alto nível de detalhamento e a MIT 1 - Baixo Detalhamento para casos de uso com baixo nível de detalhamento. A vantagem de se ter essa divisão é que o inspetor não tem que perder tempo lendo itens de verificação que não ajudarão a encontrar problemas para um determinado tipo de caso de uso.

MIT 1 - Alto detalhamento		MIT 1 - Baixo detalhamento	
<b>1AA. Visibilidade do Status do Sistema</b>		<b>1BA. Visibilidade do Status do Sistema</b>	
<b>1AA1</b>	Verifique se há algum texto no FP, FA e FE que informa em que parte do sistema o usuário se encontra;	<b>1BA1</b>	Verifique no FP, FA, FE se há alguma informação em que parte do sistema o usuário se encontra;
<b>1AA2</b>	Verifique se há algum texto no FP, FA e FE que informa ao usuário o que foi realizado após uma persistência de dados. Por exemplo: quando há alteração ou exclusão de algo, uma mensagem de texto é apresentada.	<b>1BA2</b>	Verifique se no FP, FA e FE há alguma informação para o usuário do que foi realizado após uma persistência de dados.
<b>1AB. Concordância entre o sistema e o mundo real</b>		<b>1BB. Concordância entre o sistema e o mundo real</b>	
<b>1AB1</b>	Verifique se os nomes de campos, telas, botões, links, mensagens de erros e textos informativos no FP, FA, FE e RN apresentam conceitos familiares aos usuários, ou seja, segue as convenções do mundo real;		
<b>1AB2</b>	Verifique se as opções ou campos informados pelo sistema no FP, FA e FE estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.	<b>1BB1</b>	Verifique se as opções ou campos informados pelo sistema no FP, FA e FE estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>1AC. Controle e liberdade ao usuário</b>		<b>1BC. Controle e liberdade ao usuário</b>	
<b>1AC1</b>	Verifique se o usuário, através do FA e FE, pode desfazer ou refazer algo que envolva persistência de dados no sistema. Por exemplo: pode excluir ou alterar dados inseridos.	<b>1BC1</b>	Verifique se o usuário, através do FA e FE, pode desfazer ou refazer algo que envolva persistência de dados no sistema. Por exemplo: pode excluir ou alterar dados inseridos.
<b>1AD. Consistência e padrões</b>			
<b>1AD1</b>	Verifique se todos os botões ou links possuem nomes que indicam funcionalidades diferentes no FP, FA, FE e RN, ou seja, não há botões ou links com nomes diferentes, mas com a mesma funcionalidade;		
<b>1AD2</b>	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões apresentados no FP, FA e FE;		
<b>1AD3</b>	Verifique se os nomes de campos, telas, botões e links no FP, FA, FE e RN apresentam um único sentido, sem ambiguidades.		

Figura 33 - Alto e Baixo Detalhamento da Versão 2 da MIT 1

Outros códigos como “MIT 1 pouco produtiva pois demora checar todas as heurísticas”, “É difícil relacionar problemas a uma heurística”, “Não é fácil identificar qual heurística correspondente”, “É difícil ser imparcial quanto a utilização das heurísticas”, “Não é fácil encontrar as heurísticas por causa da estrutura do documento de impressão”, “MIT é fácil de lembrar com consulta”, foram também tratados com as melhorias feitas acima.

### 5.3 Melhorias na MIT 2 v1

Um mesmo inspetor deu a “Sugestão de adaptar ao mockup os itens de avaliação que no momento são desnecessários” e a “Sugestão de uma preparação adequada dos itens”, porém estas sugestões tornariam a MIT 2 muito mais complexa. Para solucionar o problema de se ter itens de verificação desnecessários para avaliar certos tipos de *mockups* e para resolver os problemas relatados nos códigos “Há itens desnecessários por ser tratar de uma avaliação em *mockup* impresso” e “MIT 2 com heurísticas amplas mas com poucas ocorrências”, dividiu-se a MIT 2 em Alto Detalhamento e Baixo Detalhamento (Figura 34). Onde a MIT 2 - Alto Detalhamento é para inspecionar *mockups* que apresentem as mensagens do sistema (mensagens de erro, textos informativos, advertências). E a MIT 2 - Baixo Detalhamento para *mockups* que não apresentem as mensagens do sistema. A vantagem disso é que o inspetor não precisa ler itens de verificação desnecessários para o tipo de *mockup* que ele esta utilizando.

MIT 2 - Alto detalhamento		MIT 2 - Baixo detalhamento	
<b>2AA. Visibilidade do Status do Sistema</b>		<b>2BA. Visibilidade do Status do Sistema</b>	
2AA1	Verifique se há informações textuais ou nome nos <i>Mockups</i> que informa em que parte do sistema o usuário se encontra.	2BA1	Verifique se há informações textuais ou nome nos <i>Mockups</i> que informa em que parte do sistema o usuário se encontra.
2AA2	Verifique se há algum texto informativo ou mensagem que informa ao usuário o que foi realizado após uma persistência de dados (alteração, exclusão, etc).	2BA2	Verifique se há algum texto que informa ao usuário o que foi realizado após uma persistência de dados (alteração, exclusão, etc).
<b>2AB. Concordância entre o sistema e o mundo real</b>		<b>2BB. Concordância entre o sistema e o mundo real</b>	
2AB1	Verifique se os símbolos, palavras, frases, nome de botões, links, mensagens de erros e textos informativos estão expressas em uma representação facilmente compreendida pelo usuário, ou seja, possuem uma linguagem familiar;	2BB1	Verifique se os símbolos, palavras, frases, nome de botões, links e textos informativos estão expressas em uma representação facilmente compreendida pelo usuário, ou seja, possuem uma linguagem familiar;
2AB2	Verifique se as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.	2BB2	Verifique se as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>2AC. Controle e liberdade ao usuário</b>		<b>2BC. Controle e liberdade ao usuário</b>	
2AC1	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;	2BC1	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;
2AC2	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado.	2BC2	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado.
<b>2AD. Consistência e padrões</b>		<b>2BD. Consistência e padrões</b>	
2AD1	Verifique se a terminologia, gráficos e símbolos estão consistentes;	2BD1	Verifique se a terminologia, gráficos e símbolos estão consistentes;
2AD2	Verifique se há padrões adotados em relação a layout, formatação e controles. Por exemplo, botão “Salvar” no canto superior direito em um <i>Mockup</i> e em outro no canto inferior direito;	2BD2	Verifique se há padrões adotados em relação a layout, formatação e controles. Por exemplo, botão “Salvar” no canto superior direito em um <i>Mockup</i> e em outro no canto inferior direito;
2AD3	Verifique se todos os botões/links possuem nomes/ícones que indicam funcionalidades diferentes, ou seja, não há botões/links com nomes/ícones diferentes, mas com a mesma funcionalidade;	2BD3	Verifique se todos os botões/links possuem nomes/ícones que indicam funcionalidades diferentes, ou seja, não há botões/links com nomes/ícones diferentes, mas com a mesma funcionalidade;
2AD4	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões.		

Figura 34 - Alto e Baixo Detalhamento da Versão 2 da MIT 2

Além disso, um inspetor sugeriu: *“Sugestão de compreender quando cada MIT apresenta características singulares e quando for feita a inspeção em um único modelo, esta MIT possuir todas as características”*. Porém esta sugestão também tornaria a MIT 2 mais complexa. Como o objetivo da MIT 2 é ser fácil de usar, esta sugestão não foi aceita.

Na versão 2 da MIT 2 também buscou-se resumir alguns itens de verificação que estavam muito extensos, com a missão de torná-las mais objetivos e menos cansativos. Esta modificação tratou reclamações feitas pelos inspetores, como *“Sugestão de generalizar mais as heurísticas”*, *“Grande quantidade de heurísticas torna a inspeção demorada, menos produtiva e chata”*, *“MIT 2 causa fadiga e cansaço”* e *“É difícil ser imparcial quanto à utilização das heurísticas”*.

O item de verificação MIT - 2F4 *“Verifique se é fácil reconhecer/visualizar dados já fornecidos”* foi retirado da técnica, pois através do reclamação *“Necessidade de especificar melhor o item MIT - 2F4”* observou-se que este item não tem como ser verificado em *mockups*.

#### 5.4 Melhorias na MIT 3 v1

Um inspetor apontou a *“Necessidade de criar uma opção na MIT 3B de cancelar, voltar ou desfazer uma atividade”*, além de outro ter solicitado a *“Necessidade de criar uma opção na MIT 3B de cancelar uma atividade”*. Para resolver este problema, criou-se o seguinte item de verificação (Figura 35) que pertence à heurística 3C – Controle e liberdade do usuário:

<b>3C3</b>	<b>Verifique se o usuário através de alguma atividade consegue cancelar, voltar ou desfazer algo.</b>
------------	---

**Figura 35** - Item de verificação 3C3 da Versão 2 da MIT 3

Na versão 2 da MIT 3 também buscou-se tornar alguns itens de verificação mais claros. Esta modificação tratou de certa forma reclamações como *“MIT 3 possui heurísticas parecidas”*, *“Sugestão de criar heurísticas menos genéricas e mais definidas”*, *“É difícil ser imparcial quanto a escolha da heurística na hora da classificação por causa da tendência de usar determinada heurística para a maioria dos casos”* e *“MIT 3 não é tão fácil de lembrar”*.

Além disso, as reclamação *“MIT 3 não contribuiu tanto”*, *“MIT 3 não ajudou a identificar nenhum novo problema, apenas confirmou o que o inspetor já sabia ser um problema”*, *“Alguns erros ainda ficam escondidos”*, *“Alguns erros da MIT 3G podem ser diferentes dos citados”* não influenciaram em nenhuma das mudanças, pois não estavam claras ou não ajudavam a identificar um problema na técnica. A modificação solicitada no código *“Necessidade de desmembrar mais a MIT 3G”* não foi realizada porque foi observado que a MIT 3G já está bem clara e objetiva, não tendo como desmembrar mais.

## **5.5 Considerações Finais**

Este capítulo teve como objetivo apresentar as melhorias realizadas nos itens de verificação das técnicas MIT apresentando os códigos que contribuíram para o seu aprimoramento.

No próximo capítulo serão apresentadas considerações finais assim como as perspectivas de trabalhos futuros para continuação desta pesquisa.

## CAPÍTULO 6 - CONCLUSÕES E PERSPECTIVAS FUTURAS

### 6.1 Conclusões

O foco desta dissertação foi apoiar a avaliação de usabilidade de modelos de projeto de desenvolvimento de *software*. Com este propósito, foi definido o conjunto de técnicas chamada MIT (*Model Inspection Technique for Usability Evaluation*), que são técnicas de inspeção de usabilidade específica para avaliação de modelos de projeto de *software*. Estas técnicas foram desenvolvidas de forma que pudesse ser empregada pelos próprios envolvidos nos projetos de desenvolvimento de *software* (como avaliadores de requisitos) na avaliação de usabilidade de seus modelos.

As MITs foram propostas com base em uma análise de característica de técnicas de usabilidade com foco em modelos de projeto existentes na literatura e o seu desenvolvimento se apoiou em um estudo experimental realizado na academia. Até o presente momento, as MITs foram avaliadas experimentalmente através um estudo *in vitro* (estudos de viabilidade). A seguir são apresentados os requisitos que guiaram a proposta das técnicas e os resultados relacionados, medidos através do estudo experimental executado:

- Ser fácil de aprender e de utilizar: O tempo de treinamento da Avaliação Heurística para todos os participantes foi de 1h e o tempo de apresentação das técnicas MITs foi de 15min para o grupo da MIT. Logo após o treinamento todos os participantes foram capazes de realizar a detecção de problemas de usabilidade;

- Apresentar bom nível de eficácia (razão entre o número de defeitos detectados e o número total de defeitos existentes): A eficácia média do grupo de inspetores que utilizou a MIT 1 foi de 17% enquanto eficácia média utilizando AH foi de 11%. Já a eficácia média do grupo de inspetores que utilizou a MIT 2 foi de 16% enquanto eficácia média utilizando AH foi de 14%. E por fim a eficácia média do grupo de inspetores que utilizou a MIT 3 foi de 129% enquanto eficácia média utilizando AH foi de 21%;

- Apresentar bom nível de eficiência (razão entre o número de defeitos por tempo de inspeção): a eficiência média da inspeção utilizando a MIT 1 foi de 11,35 defeitos/hora, enquanto a AH mostrou uma eficiência de 8,62 defeitos por hora. Já a MIT 2, mostrou uma eficiência de 13,78 defeitos por hora, enquanto a AH mostrou eficiência de

13,80 defeitos por hora. E por fim a MIT 3 apresentou uma eficiência de 12,32 defeitos por hora, enquanto a AH mostrou eficiência de 7,64 defeitos por hora.

- Oferecer uma boa relação custo-benefício na sua aplicação: o esforço nas etapas de detecção e discriminação, componente de maior custo nas inspeções, foi medido no estudo *in vitro*. Neste estudo, o esforço médio:

- Na MIT 1: somando o esforço na atividade de detecção (1 hora e 30 minutos) e discriminação (2 hora e 30 minutos) foi de 4 horas.
- Na MIT 2: somando o esforço na atividade de detecção (46 minutos) e discriminação (1 hora e 35 minutos) foi de 2 horas e 21 minutos.
- Na MIT 3: somando o esforço na atividade de detecção (38 minutos) e discriminação (1 hora e 15 minutos) foi de 1 hora e 53 minutos.

O custo de capacitação neste estudo foi de 1 hora de treinamento sobre usabilidade e 15 minutos de apresentação das técnicas MITs, totalizando 1 hora e 15 minutos de apresentações e não foi necessário contratar um especialista para realizar a inspeção. Desta forma, foi possível realizar a inspeção usando as três MITs investindo 9 horas e 29 minutos dos analistas de sistemas e suporte disponibilizados para esta atividade.

Ao se analisar estes resultados, deve-se considerar as diferentes ameaças à validade do estudo. Os resultados do estudo realizado proporcionam indícios de que as MITs apresentam bom nível de eficácia, eficiência, consegue ser aplicada após um único treinamento (com duração média de 1 hora e 30 minutos) e seus custos de aplicação são baixos. Além disso, os inspetores que utilizaram as MITs apontaram um número maior de problemas de usabilidade. A utilização das MITs proporciona direcionamento para a execução da inspeção de usabilidade, tornando a inspeção mais fácil para os inspetores menos experientes. Como desvantagem de utilização da MIT 2, tem-se o maior tempo de inspeção.

## 6.2 Contribuições

As principais contribuições desta tese são:

- Desenvolvimento de tecnologias de apoio à inspeção de usabilidade em modelos:
  - Definição de uma técnica de inspeção de usabilidade específica para modelos de projeto de *software*, que pode ser empregada pelos próprios envolvidos nos projetos de desenvolvimento de *software*, presente no Apêndice B;

- o Elaboração de recursos de apoio à realização de inspeções de usabilidade de modelos em organizações de desenvolvimento de *software*: instruções para a inspeção de usabilidade com a AH e com as MITs (Apêndice C) e planilhas para relato de discrepâncias (Apêndice D).
- Estudo da área de Usabilidade de Modelos de Projeto de *software*, voltado para questões características de avaliação de modelos que garantem a usabilidade:
  - o Realização de um mapeamento sistemático para caracterizar os métodos utilizados na avaliação de usabilidade em modelos de análise e projeto, o que implicou em:
    - ✓ Definição de um protocolo para realização de mapeamento sistemático que pode ser utilizado/estendido em novas pesquisas sobre este tema;
    - ✓ Estabelecimento de uma base de conhecimento sobre métodos avaliação de usabilidade em modelos de análise e projeto, que foi utilizada nesta pesquisa e pode contribuir para outras pesquisas da área.
- Definição de pacote para planejamento, execução e análise de estudo de viabilidade que permitiu:
  - o Avaliação e evolução da técnica e dos recursos de apoio à inspeção de usabilidade;
  - o Disseminação do conhecimento sobre realização de estudo experimental, incluindo análise qualitativa tendo por base os procedimentos do método *Grounded Theory* (STRAUSS e CORBIN 1998) e avaliação da aceitação de tecnologia, baseada no modelo TAM (*Technology Acceptance Model*) (DAVIS 1989).

### 6.3 Perspectivas Futuras

A realização desta pesquisa resultou no desenvolvimento de uma técnica de leitura para inspeção de usabilidade específica para modelos de análise e projeto, que pode ser utilizada pelos próprios *stakeholders* dos projetos de desenvolvimento de *software* na avaliação de usabilidade de seus modelos. Estes resultados abrem novas perspectivas de pesquisa,

que podem ser exploradas em trabalhos futuros. Alguns desses trabalhos futuros são detalhados a seguir.

### **6.3.1 Evolução do Mapeamento Sistemático**

O mapeamento sistemático desta pesquisa abrangeu os anos de 2000 a 2011. Como novos estudos relacionados ao tema desta pesquisa são realizados constantemente, faz-se necessário realizar a evolução deste mapeamento compreendendo os anos posteriores ao mapeamento realizado.

### **6.3.2 Realização de um Novo Estudo de Viabilidade**

O estudo de viabilidade descrito nesta dissertação contou com um número pequeno de participantes. Este estudo poderia ser repetido com um número maior de participantes utilizando a versão 2 das técnicas, de forma a aumentar a significância dos resultados obtidos e reduzir as ameaças à conclusão do estudo, além de coletar mais dados qualitativos para esta pesquisa.

Outro ponto que precisa ser melhorado no segundo estudo de viabilidade é o Diagrama de Atividades (DA), pois o DA do primeiro estudo foi realizado pela autora da técnica, e portanto não era um DA pertencente à um ambiente real.

Além disso, um novo estudo comparando as técnicas MITs com outras técnicas de inspeção específicas para os modelos poderia ser realizado.

### **6.3.3 Realização de Estudo de Observação**

Segundo SHULL *et al.* (2001), um estudo de observação deve ser realizado para aprimorar o entendimento dos pesquisadores em relação à aplicação da tecnologia e possibilitar seu refinamento. Este estudo visa avaliar as etapas do processo para garantir que cada uma é eficaz e que a ordem em que são executadas faz sentido.

Para avaliar uma técnica de leitura, é essencial que aquele que a utilize siga as instruções passo a passo, pelo menos no início do processo. O estudo de observação também ajudará a avaliar se a ordem de aplicação proposta atualmente é a melhor para a avaliação com as MITs. Para isto, é necessário montar uma infra-estrutura para a realização do estudo, por exemplo, um laboratório de observação. Também é preciso compatibilizar o tempo entre os inspetores e os pesquisadores que atuarão como observadores.

#### **6.3.4 Realização de Estudo de Caso na Indústria**

SHULL *et al.* (2001) apontam que estudos de caso na Indústria devem ser realizados para identificar se existem problemas de integração na aplicação da tecnologia proposta em um ambiente industrial.

Com a realização deste estudo, podem-se observar indícios sobre a efetividade das técnicas MITs. O seu propósito é: (1) caracterizar a aplicação da técnica no contexto de um ciclo de vida de desenvolvimento e (2) identificar se a aplicação da tecnologia possui alguma interação negativa com o ambiente industrial.

## REFERÊNCIAS BIBLIOGRÁFICAS

- ABRAHÃO, S., CONDORI-FERNANDEZ, N., OLSINA, L., PASTOR, O., 2003. "Defining and Validating Metrics for Navigational Models". In: Proceedings of the 9th International Symposium on Software Metrics, Sydney, p. 200 - 210.
- ABRAHÃO, S., INSFRAN, E., 2006. "Early Usability Evaluation in Model Driven Architecture Environments". In: Proceedings of 6th International Conference on Quality Software, Beijing, p. 287 - 294.
- ANDERSON, J. R., 1983. "The architecture of cognition. Cambridge", MA: Harvard University Press.
- ATTERER, R., 2005. "Where Web Engineering Tool Support Ends: Building Usable Websites". In Proceedings of the 20th Annual ACM Symposium on Applied Computing, Santa Fe, New Mexico, p. 12–17.
- ATTERER, R., SCHMIDT, A., 2005. "Adding Usability to Web Engineering Models and Tools". In Proceedings of the 5th International Conference on Web Engineering, Sydney, v. 3579, p. 185 - 211.
- ATTERER, R., SCHMIDT, A., HUSSMANN, H., 2006. "Extending Web Engineering Models and Tools for Automatic Usability Validation". In: Journal of Web Engineering, v. 5, issue 1, p. 43-64.
- ATTERER, R., 2008. "Model-Based Automatic Usability Validation – a Tool Concept for Improving Web-Based UIs". In: Proceedings of the 5th Nordic conference on Human-computer interaction: building bridges, New York, p. 13 - 22.
- BELGAMO, A., FABBRI, S., MALDONADO, J., 2005. "Avaliando a Qualidade da Técnica GUCCRA com Técnica de Inspeção." In: VIII Workshop on Requirements Engineering, Porto, v. 1, p. 1 - 10.
- BLACKMON, M. H., POLSON, P. G., KITAJIMA, M., LEWIS, C., 2002. "Cognitive walkthrough for the web". In: Proceedings of the SIGCHI conference on Human factors in computing systems, Minneapolis, p. 463 – 470.
- BOLCHINI, D., GARZOTTO, F., 2007. "Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+". In: Workshop on Web Information Systems Engineering, Nancy, v. 4832, p. 481 - 492.
- BONIFÁCIO, B., FERNANDES, P., OLIVEIRA, H. A. B. F., CONTE, T., 2011. "UBICUA: A customizable usability inspection approach for web mobile applications". In Proceedings of the IADIS International Conference on WWW/Internet.
- CARD, S., MORAN, T. & NEWELL, A., 1983. "The Psychology of Human-Computer Interaction". Hillsdale, New Jersey: Erlbaum.
- CARVER, J., JACCHERI, L., MORASCA, S., SHULL, F., 2003. "Issues in Using Students in Empirical Studies in Software Engineering Education". In Proceedings of the 9th International Symposium on Software Metrics, Sydney, Australia, p. 239 – 249.
- CONTE, T., MASSOLAR, J., MENDES, E., TRAVASSOS, G. H., 2007. "Usability Evaluation based on Web Design Perspectives". In Proceedings of the First International Symposium on Empirical Software Engineering and Measurement, p. 146 – 155.
- CONTE, T., 2009. "Técnica de Inspeção de Usabilidade Baseada em Perspectivas de Projeto Web". 194 pages. Tese (Doutorado) - Universidade Federal do Rio de Janeiro. Rio de Janeiro.

- CONTE, T., MASSOLAR, J., MENDES, E., TRAVASSOS, G. H., 2009. "Web usability inspection technique based on design perspectives". In: IET Software, v. 3, issue 2, p. 106 – 123.
- COSTA NETO, M. A., SOUZA, A. J., LEITE, J. C., 2011. "Identificando problemas de usabilidade através de inspeção no modelo de interação". In Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, p. 124 - 133.
- DAVIS, F., 1989. "Perceived usefulness, perceived ease of use, and user acceptance of information technology". MIS Quarterly, v. 13, n. 3, p. 319 - 339.
- DE SOUZA, C. S., LEITAO, C. F., PRATES, R. O., DA SILVA, E. J., 2006. "The Semiotic Inspection Method". In Proceedings of VII Brazilian Symposium on Human Factors in Computing Systems, p. 148 - 157.
- DONAHUE, G. M., 2001. "Usability and the Bottom Line". In: Journal of IEEE Software, v. 18, issue 1, p. 31 - 37.
- FERNANDEZ, A., 2009a. "WUEP: Un Proceso de Evaluación de Usabilidad Web Integrado en el Desarrollo de Software Dirigido por Modelos". 173 pages. Dissertação (Mestrado) - Universitat Politècnica de València.
- FERNANDEZ, A., INSFRAN, E., ABRAHÃO, S., 2009b. "Integrating a Usability Model into Model-Driven Web Development Processes". In: Proceedings of the 10th International Conference on Web Information Systems Engineering, Poznan, v. 5802, p. 497 - 510.
- FERNANDEZ, A., ABRAHÃO, S., INSFRAN, E., 2010. "Towards to the validation of a usability evaluation method for model-driven web development". In: Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement, Bolzano-Bozen, issue 54.
- FERNANDEZ, A., ABRAHÃO, S., INSFRAN, E., 2011a. "A Web Usability Evaluation Process for Model-Driven Web Development". In: Proceedings of the 23rd international conference on Advanced information systems engineering, v. 6741, p. 108 - 122.
- FERNANDEZ, A., INSFRAN, E., ABRAHÃO, S., 2011b. "Usability evaluation methods for the web: A systematic mapping study". In: Journal of Information and Software Technology, v. 53, issue 8, p. 789 - 817.
- FERRÉ, X., JURISTO, N., WINDL, H., CONSTANTINE, L., 2001. "Usability Basics for Software Developers". In: Journal of IEEE Software, v. 18, issue 1, p. 22 - 29.
- FERRÉ, X., JURISTO, N., MORENO, A., 2004. "Improving Software Engineering Practice with HCI Aspects". In: Proceedings of International Conference on Software Engineering Research and Applications, Los Angeles, v. 3026, p. 349 - 363.
- FERRÉ, X., JURISTO, N., MORENO, A., 2005. "Framework for Integrating Usability Practices into the Software Process". In: Proceedings of the 6th International Conference Product Focused Software Process Improvement, v. 3574, p. 202 - 215.
- FERREIRA, D. S., OLIVEIRA, K. M. A., ALENCAR, A. A. "Inclusão de Usabilidade no Processo de Desenvolvimento de Software da DATAPREV". In Proceedings of the IX Brazilian Symposium on Human Factors in Computing Systems, p. 263 - 264.
- GÓMEZ, J., CACHERO, C., PASTOR, O., 2001 "Conceptual Modeling of Device-Independent Web Applications: Towards a Web Engineering Approach". In: Journal of IEEE MultiMedia, Los Alamitos, v. 7, issue 2, p. 26 - 39.
- GRISLIN, E, KOLSKI, C, 1996. "Evaluation des interfaces homme-machine lors du développement de système interactif". In Journal of Technique et Science Informatiques, Valenciennes, v. 15, issue 3, p. 265 - 296.

- HITZ, M., LEITNER, G., MELCHER, R., 2006. "Usability of Web Applications". In: Web Engineering: The Discipline of Systematic Development of Web Applications, Chapter 11.
- HORNBAEK, K., HØEGH, R. T., PEDERSEN, M. B., STAGE, J., 2007. "Use Case Evaluation (UCE): A Method for Early Usability Evaluation in Software Development". In: Proceedings of the 11th International Conference on Human-Computer Interaction, Rio de Janeiro, v. 4662, p. 578 - 591.
- ISO/IEC (1998). ISO 9241-11: Ergonomic Requirements for Office Work with Visual Display Terminals, Geneva.
- ISO/IEC (2002). ISO/IEC Standard 12207:1995: International Standard: Information Technology. Software Life Cycle Processes. Amendment 1. Amd.1:2002. ISO, Geneva, Switzerland.
- JALALI, S. e WOHLIN, C., 2012. "Systematic literature studies: database searches vs. backward snowballing". In Proceedings of the ACM-IEEE international symposium on Empirical software engineering and measurement, p 29 - 38.
- JURISTO, N., MORENO, A., SÁNCHEZ, M., 2004. "Clarifying the Relationship between Software Architecture and Usability". In: Proceedings of the 6th International Conference on Software Engineering and Knowledge Engineering, Banff, p. 378 - 383.
- JURISTO, N., MORENO, A., SÁNCHEZ, M., BARANAUSKAS, M., 2007a. "A Glass Box Design: Making the Impact of Usability on Software Development Visible". In: Proceedings of the 11th International Conference on Human-Computer Interaction, Rio de Janeiro, v. 4663, p. 541 - 554.
- JURISTO, N., MORENO, A., SÁNCHEZ, M., 2007b. "Guidelines for Eliciting Usability Functionalities". In: Journal of IEEE Transactions on Software Engineering, Piscataway, v. 33, issue 11, p. 744 - 758.
- KAPPEL, G., PRÖLL, B., REICH, S., RETSCHITZEGGER, W., 2006a. "An Introduction to Web Engineering". In: Kappel, G., Pröll, B., Reich, S., Retschitzegger, W. (eds), Web Engineering: The Discipline of Systematic Development of Web Applications, John Wiley & Sons.
- KAPPEL, G., PRÖLL, B., REICH, S., RETSCHITZEGGER, W., 2006b. "An Introduction to Web Engineering". In: Web Engineering - The Discipline of Systematic Development of Web Applications, 366 pages.
- KIERAS, D. E., 2009 "Model-based evaluation", In: Sears, A. and Jacko, J. (Eds.), The Human-Computer Interaction: Development Process, p. 294 - 310.
- KITCHENHAM, B., 2004. "Procedures for Performing Systematic Reviews". In: Keele technical report SE0401 and NICTA technical report 0400011T.1.
- KITCHENHAM, B., DYBA, T., JORGENSEN, M., 2010. "The value of mapping studies – A participant-observer case study". In: Proceedings of the 14th International Conference on Evaluation and Assessment in Software Engineering, Keele, p. 1 - 9.
- KOCH, N., KRAUS, A., 2002. "The Expressive Power of UML-based Web Engineering". In: Proceedings of the Second International Workshop on Web-oriented Software Technology, Malaga.
- LAITENBERGER, O., DREYER, H. M., 1998. "Evaluating the usefulness and the ease of use of a web-based inspection data collection tool". In Proceedings of the 5th International Symposium on Software Metrics (1998), page 122.
- LAUGHERY, K. R., 1989. "Micro SAINT - A tool for modeling human performance in systems". In G.R. McMillan, D. Beevis, E. Salas, M.H. Strub, R. Sutton, & L. Van Breda (Eds.), Applications of human performance models to system design. New York: Plenum Press, p. 219 - 230.

- LEITE, J. C. S. D. P., DOORN, J. H., HADAD, G. D. S., KAPLAN, G. N., 2005. "Scenario inspections". In: *Journal of Requirements Engineering*, New York, v. 10, issue 1, p. 1 - 21.
- LIANG, L., DENG, X., 2009. "A Collaborative Task Modeling Approach Based on Extended GOMS". In: *Proceedings of the 2009 International Conference on Electronic Computer Technology*, Macau, p. 375 - 378.
- LUNA, E. R., PANACH, J. I., GRIGERA, J., ROSSI, G., PASTOR, O., 2010. "Incorporating usability requirements in a test/model-driven web engineering approach". In: *Journal of Web Engineering*, v. 9, issue 2, p. 132 - 156.
- MAFRA, S., BARCELOS, R., TRAVASSOS, G. H., 2006. "Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software". In: *Proceedings of 20th Brazilian Symposium on Software Engineering*, v.1, p. 239 - 254.
- MALLARDO, T., CALEFATO, F., 2003. "Tool support for Geographically Dispersed Inspection Teams, Software Process Improvement and Practice", v. 8: p. 217-231.
- MANN, H. B., WHITNEY, D. R., 1947. "On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other". In *Annals of Mathematical Statistics* 18, p. 50 - 60.
- MASSOLAR, J., 2008. "Uma Abordagem baseada na Engenharia para Desenvolvimento e Garantia da Qualidade de Aplicações Web". Exame de Qualificação - Universidade Federal do Rio de Janeiro.
- MATERA, M., COSTABILE, M. F., GARZOTTO, F., PAOLINI, P., 2002. "SUE Inspection: An Effective Method for Systematic Usability Evaluation of Hypermedia". In: *IEEE Transactions on Systems, Man and Cybernetics, Part A*, v. 32, issue 1, p. 93 - 103.
- MATERA, M., RIZZO, F., CARUGHI, G. T., 2006. "Web Usability: Principles and Evaluation Methods". In: *Journal of Web Engineering*, New York, Chapter 5, p. 143 - 180.
- MAYHEW, D. J., 1999. "The usability engineering lifecycle: a practitioner's handbook for user interface design". In Morgan Kaufmann Publishers Inc., San Francisco, CA.
- MENDES, E., 2005. "A systematic review of Web engineering research". In: *Proceedings of International Symposium on Empirical Software Engineering*, p. 498 - 507.
- MOLINA, F., TOVAL, A., 2009. "Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems". In: *Journal Advances in Engineering Software*, Oxford, v. 40, issue 12, p. 1306 - 1317.
- MORAGA, M. A., CALERO, C., PIATTINI, M., and DIAZ, O., 2007. "Improving a portlet usability model. Software Quality Control". In: *Journal Software Quality Control*, Hingham, v. 15, issue 2, p. 155 - 177.
- NAMAHN 2001. "Using eye tracking for usability testing. a research note by Namahn".
- NIELSEN, J., 1994a. "Guerrilla HCI: using discount usability engineering to penetrate the intimidation barrier". Orlando.
- NIELSEN, J., 1994b. "Heuristic evaluation". In: Jakob Nielsen, Mack, R. L. (eds), *Usability inspection methods, Heuristic Evaluation*, New York.
- OFFUTT, J., 2002. "Quality Attributes of Web Software Applications". In: *Journal of IEEE Software*, v. 19, issue 2, p. 25 - 32.
- OLSINA, L., ROSSI, G., 2002. "Measuring Web Application Quality with WebQEM". In: *Journal of IEEE Multimedia*, v. 9, issue 4, p. 20 - 29.
- OLSINA, L., COVELLA, G., ROSSI, G., 2006. "Web Quality". In: *Journal of Web Engineering*, Chapter 4.
- PANACH, J. I., VALVERDE, F., PASTOR, O., 2007. "Improvement of a Web Engineering Method through Usability Patterns". In: *Proceedings of the 2007*

- international conference on Web information systems engineering, v. 4832, p. 441 - 446.
- PASTOR, O., 2004. "Fitting the Pieces of the Web Engineering Puzzle". In: Proceedings of 18th Brazilian Symposium on Software Engineering, Brasília, v. 1, p. 10 - 22.
- PFLEEGER, S. L., 1999. "Albert Einstein and Empirical Software Engineering." IEEE Computer, v. 32, n. 10, p. 32-38.
- PRATES, R. O., BARBOSA, S. D. J., 2003. "Avaliação de Interfaces de Usuário – Conceitos e Métodos". In: Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação, Campinas, Chapter 6, v. 2, p. 245 - 293.
- PRATES, R. O., BARBOSA, S. D. J., 2007. "Introdução à Teoria e Prática da Interação Humano-Computador fundamentada na Engenharia Semiótica". In: Proceedings of XXVII Congresso da Sociedade Brasileira de Computação - Jornadas de Atualização em Informática, p. 263 - 326.
- POLSON, P., LEWIS, C., RIEMAN, J., WHARTON, C., 1992. "Cognitive walkthroughs: a method for theory-based evaluation of user interfaces". International Journal of Man-Machine Studies, v. 36, n. 5, p. 741-773.
- ROCHA, H. V., BARANAUSKAS, M. C. C., 2003. "Design e Avaliação de Interfaces Humano-Computador". NIED, UNICAMP, Campinas.
- SEFFAH, A., DJOUAB, R., ANTUNES, H., 2011. "Comparing and Reconciling Usability-Centered and Use Case-Driven Requirements Engineering Processes". In: Proceedings of the 2nd Australasian conference on User interface, Los Alamitos, p. 132 – 139.
- SILVA, T. S., SILVEIRA, M. S., 2008. "Antecipando a avaliação de IHC: verificação de diretrizes a partir de modelos". In Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems, p. 248 - 251.
- SILVA, T. S., SILVEIRA, M. S., 2010. "Validação de um Método para Identificação de Problemas de Usabilidade a partir de Diagramas UML". In Proceedings of the IX Brazilian Symposium on Human Factors in Computing Systems, p. 179 - 188.
- SHULL, F., CARVER, J., TRAVASSOS, G. H., 2001. "An empirical methodology for introducing software processes". In 9th ACM SIGSOFT international symposium on Foundations of software engineering, p. 288 - 296.
- SHULL, F., MENDONÇA, M. G., BASILI, V., CARVER, J., MALDONADO, J. C., FABBRI, S., TRAVASSOS, G. H., FERREIRA, M. C., 2004. "Knowledge-Sharing Issues in Experimental Software Engineering". In: Journal of Empirical Software Engineering, Hingham, v. 9, issue 1-2, p. 111 - 137.
- SOMMARIVA, L., BENNITI, F. B. V., DALCIN, F. S. "UsabilityGame: Jogo Simulador para Apoio ao Ensino de Usabilidade". In Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, p. 61 - 65.
- SPSS, 2011. SPSS version 20.0 for Windows. SPSS Inc.
- STRAUSS, A., CORBIN, J., 1998. "Basics of Qualitative Research: Techniques and Procedures for Developing Grounded Theory". 2 ed. London, SAGE Publications.
- TAO, Y., 2005. "Work in Progress - Introducing Usability Concepts in Early Phases of Software Development". In: Proceedings 35th Annual Conference Frontiers in Education, Indianapolis, p. T4C - 7.
- THIEL, M., PETTER, A., BEHRING, A., 2007. "Usability Aware Model Driven Development of User Interfaces", p. 83 - 92.
- TRAVASSOS, G. H., SHULL, F., FREDERICKS, M., BASILI, V., 1999. "Detecting defects in object-oriented designs: using reading techniques to increase software quality". In: Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, v. 34, n. 10, p. 47 - 56.

- TRAVASSOS, G. H., SHULL, F., CARVER, J., BASILI, V., 2002. "Reading Techniques for OO Design Inspections". University of Maryland.
- TRAVASSOS, G. H., BARROS, M., 2003. "Contributions of In Virtuo and In Silico Experiments for the Future of Empirical Studies in Software Engineering". In: Proceedings of the 2nd Workshop in Workshop Series on Empirical Software Engineering, Rome, p. 117 - 130.
- TRIACCA, L., INVERSINI, A., BOLCHINI, D., 2005. "Evaluating Web usability with MiLE+". In: Proceedings of the Seventh IEEE International Symposium on Web Site Evolution, p. 22 - 29.
- VALENTIM, N. M. C., CONTE, T., 2011. "Mapeamento Sistemático para a geração da Técnica de Inspeção de Modelos para garantir a Usabilidade". In: USES Technical Report Número RT-USES-2011-0006, Manaus.
- VALENTIM, N. M. C., CONTE, T., 2012. "Proposta Inicial do Conjunto de Técnicas MIT". In USES Technical Report Número RT-USES-2012-0002, Manaus.
- VALENTIM, N. M. C., OLIVEIRA, K. M., CONTE, T., 2012a. "Extensão do Mapeamento Sistemático para a geração da Técnica de Inspeção de Modelos para garantir a Usabilidade". In USES Technical Report Número RT-USES-2012-0004 (2012), Manaus.
- VALENTIM, N. M. C., OLIVEIRA, K. M., CONTE, T., 2012b. "Definindo uma Abordagem para Inspeção de Usabilidade em Modelos de Projeto por meio de Experimentação". In: XI Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC 2012), Cuiabá – MT, p. 165 - 174.
- VASCONCELOS, L. G., BALDOCHI JÚNIOR, L. A., 2011. "USABILICS: avaliação remota de usabilidade e métricas baseadas na análise de tarefas". In Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, p. 303 - 312.
- VILLEGAS, M. L., GIRALDO, W. J., GRANOLLERS, T., TREFFTZ, H. "Definición de un Marco de Desarrollo de Sistemas Interactivos Usables: a partir de la Integración de CIAF y MPIu+a". In Proceedings of the 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, p. 105 - 113.
- WINCKLER, M.; PIMENTA, M., 2002. "Avaliação de Usabilidade de Sites Web". In: Nedel, Luciana (Org.) X Escola de Informática da SBC-Sul, Caxias do Sul, p. 85 - 137.
- WÖHLIN, C., RUNESON, P., HÖST, M., OHLSSON, M. C., REGNELL, B., WESSL, A., 2000. "Experimentation in software engineering: an introduction". Kluwer Academic Publishers.
- YU, E. 1997 "Towards Modelling and Reasoning Support for Early-Phase Requirements Engineering", In: Proceedings of the 3rd IEEE International Symposium on Requirements Engineering, p. 226 - 235.

## APÊNDICE A – QUESTIONÁRIO DE AVALIAÇÃO DAS MITS UTILIZADO NO ESTUDO DE VIABILIDADE

Por gentileza, responda as questões a seguir considerando sua experiência durante a inspeção utilizando a técnica MIT:

- Em relação à sua percepção sobre a facilidade de uso da técnica MIT, qual o seu grau de concordância em relação às seguintes afirmações:

	Concordo Totalmente (100%)	Concordo Amplamente (99% - 70%)	Concordo Parcialmente (69% - 51%)	Discordo Parcialmente (50% - 31%)	Discordo Amplamente (30% - 1%)	Discordo Totalmente (0%)
Foi fácil aprender a utilizar a MIT						
Consegui utilizar a MIT da forma que eu queria						
Eu entendia o que acontecia na minha interação com a MIT						
Foi fácil ganhar habilidade no uso da MIT						
É fácil lembrar como utilizar a MIT para realizar uma inspeção de usabilidade						
Considero a MIT fácil de usar						

Comentários (opcional):
-------------------------

- Em relação à sua percepção sobre a utilidade da técnica MIT, qual o seu grau de concordância em relação às afirmações abaixo:

	Concordo Totalmente (100%)	Concordo Amplamente (99% - 70%)	Concordo Parcialmente (69% - 51%)	Discordo Parcialmente (50% - 31%)	Discordo Amplamente (30% - 1%)	Discordo Totalmente (0%)
A MIT me permitiu detectar defeitos mais rápido						
Usar a MIT melhorou o meu desempenho na inspeção (acredito ter						

encontrado um número maior de defeitos do que encontraria sem utilizar a MIT)						
Usar a MIT facilitou a inspeção						
Eu considero a MIT útil para inspeções de usabilidade						

Comentários (opcional):

3. Em relação a algumas questões específicas da técnica MIT, qual o seu grau de concordância em relação às afirmações abaixo:

	Concordo Totalmente (100%)	Concordo Amplamente (99% - 70%)	Concordo Parcialmente (69% - 51%)	Discordo Parcialmente (50% - 31%)	Discordo Amplamente (30% - 1%)	Discordo Totalmente (0%)
As heurísticas da técnica MIT são fáceis de compreender						

Comentários (opcional):

4. Use o espaço a seguir para comentários gerais que julgar necessários sobre a técnica MIT, sua aplicação, dificuldades encontradas, etc.

## APÊNDICE B – MIT V2

<b>MIT 1 – Avaliação de Usabilidade em Casos de Uso</b>	
<b>Objetivo:</b>	Verificar a usabilidade em Casos de Uso
<b>Instruções:</b>	<p>1. Antes de começar a inspeção, observe as seguintes diretrizes:</p> <ol style="list-style-type: none"> <li>a. Caso o Caso de Uso possua <b>alto nível de detalhamento</b> (como mensagens de erro, textos informativos, advertências, nome de telas, nome de campos, nome de botões, nome de links, nome de opções, entre outros), comece a inspeção utilizando as heurísticas da <b>MIT1 - Alto Detalhamento</b>.</li> <li>b. Caso o Caso de Uso possua <b>baixo nível de detalhamento</b> (ou seja, não descreve as mensagens de erro, textos informativos, advertências, nome de telas, nome de campos, nome de botões, nome de links, nome de opções, entre outros), comece a inspeção utilizando as heurísticas da <b>MIT1 - Baixo Detalhamento</b>.</li> <li>c. Caso o Caso de Uso possua <b>nível médio de detalhamento</b> (com apenas <b>alguns</b> dos itens citados acima), utilize as heurísticas da <b>MIT1 - Alto Detalhamento</b>.</li> </ol> <p>2. Para cada Caso de Uso a ser inspecionado observe e anote em qual FP<sup>11</sup>, FA<sup>12</sup>, FE<sup>13</sup> ou RN<sup>14</sup> você identifica um problema de usabilidade seguindo as heurísticas abaixo:</p>
<b>MIT1 - Alto Detalhamento</b>	
<b>1AA. Visibilidade do Status do Sistema</b>	
<b>1AA1</b>	Verifique se há algum texto no FP, FA e FE que informa em que parte do sistema o usuário se encontra;
<b>1AA2</b>	Verifique se há algum texto no FP, FA e FE que informa ao usuário o que foi realizado após uma persistência de dados. Por exemplo: quando há alteração ou exclusão de algo, uma mensagem de texto é apresentada.
<b>1AB. Concordância entre o sistema e o mundo real</b>	
<b>1AB1</b>	Verifique se os nomes de campos, telas, botões, links, mensagens de erros e textos informativos no FP, FA, FE e RN apresentam conceitos familiares aos usuários, ou seja, segue as convenções do mundo real;
<b>1AB2</b>	Verifique se as opções ou campos informados pelo sistema no FP, FA e FE estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>1AC. Controle e liberdade ao usuário</b>	
<b>1AC1</b>	Verifique se o usuário, através do FA e FE, pode desfazer ou refazer algo que

<sup>11</sup> FP: Fluxo Principal

<sup>12</sup> FA: Fluxo Alternativo

<sup>13</sup> FE: Fluxo Exceção

<sup>14</sup> RN: Regras de Negócio

	envolva persistência de dados no sistema. Por exemplo: pode excluir ou alterar dados inseridos.
<b>1AD. Consistência e padrões</b>	
<b>1AD1</b>	Verifique se todos os botões ou links possuem nomes que indicam funcionalidades diferentes no FP, FA, FE e RN, ou seja, não há botões ou links com nomes diferentes, mas com a mesma funcionalidade;
<b>1AD2</b>	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões apresentados no FP, FA e FE;
<b>1AD3</b>	Verifique se os nomes de campos, telas, botões e links no FP, FA, FE e RN apresentam um único sentido, sem ambiguidades.
<b>1AE. Prevenção de Erros</b>	
<b>1AE1</b>	Verifique se há alguma advertência do sistema que alerta o usuário através de mensagens ou textos informativos de que o que ele está fazendo pode ser inapropriado naquele momento no FP, FA, FE;
<b>1AE2</b>	Verifique se todas as opções, botões e links disponíveis possuem nomes que definem claramente que resultados ou estados serão atingidos no FP, FA, FE e RN.
<b>1AE3</b>	Verifique se os dados ou opções obrigatórios FP <sup>15</sup> , FA <sup>16</sup> , FE <sup>17</sup> e RN <sup>18</sup> estão claramente definidos;
<b>1AF. Reconhecer ao invés de lembrar</b>	
<b>1AF1</b>	Verifique se os nomes das opções, campos, telas e links são informados de forma que o usuário não tenha que se lembrar quais são eles no FP, FA, FE e RN.
<b>1AG. Flexibilidade e eficiência de uso</b>	
<b>1AG1</b>	Verifique se há nomes de links, nome de telas ou nome de campos abreviados para usuários mais experientes no FP, FA, FE e RN;
<b>1AG2</b>	Verifique se há como passos serem realizados de forma mais rápida e eficiente para usuários mais experientes no FP, FA, FE e RN;
<b>1AG3</b>	Verifique se o sistema permite que o usuário realize o que ele deseja com facilidade através dos passos do FA e FE;
<b>1AG4</b>	Verifique se o sistema permite ao usuário formas de acesso às tarefas principais no FA e FE. Por exemplo: estando no FA ou FE o usuário consegue voltar ao FP quando for viável;
<b>1AG5</b>	Verifique se as formas de acesso oferecidas pelo sistema no FA e FE diminuem o esforço de ações físicas dos usuários.
<b>1AH. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>1AH1</b>	Verifique se as mensagens de erros utilizam linguagem simples de ser entendida pelos usuários, ou seja, de acordo com o perfil do usuário no FP, FA e FE. Por

<sup>15</sup> FP: Fluxo Principal

<sup>16</sup> FA: Fluxo Alternativo

<sup>17</sup> FE: Fluxo Exceção

<sup>18</sup> RN: Regras de Negócio

	exemplo: mensagens de erro que não tenha códigos;
<b>A1H2</b>	Verifique se todas as mensagens de erros não culpam ou intimidam o usuário no FP, FA, FE;
<b>A1H3</b>	Verifique se as mensagens de erro ajudam o usuário a corrigir o erro no FP, FA e FE. Por exemplo: mensagens com indicação de recuperação;
<b>A1H4</b>	Verifique se o sistema mostra como acessar as soluções alternativas que são apresentadas em algumas mensagens de erro no FP, FA e FE.
<b>1AI. Proporcionar funções que são úteis para o usuário</b>	
<b>1AI1</b>	Verifique se todo botão ou link essencial está presente no FP, FA, FE e RN.
<b>1AI2</b>	Verifique se todo passo, botão ou link presente é útil no FP, FA, FE e RN.
<b>MIT1 - Baixo Detalhamento</b>	
<b>1BA. Visibilidade do Status do Sistema</b>	
<b>1BA1</b>	Verifique no FP <sup>19</sup> , FA <sup>20</sup> , FE <sup>21</sup> se há alguma informação em que parte do sistema o usuário se encontra;
<b>1BA2</b>	Verifique se no FP, FA e FE há alguma informação para o usuário do que foi realizado após uma persistência de dados.
<b>1BB. Concordância entre o sistema e o mundo real</b>	
<b>1BB1</b>	Verifique se as opções ou campos informados pelo sistema no FP, FA e FE estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>1BC. Controle e liberdade ao usuário</b>	
<b>1BC1</b>	Verifique se o usuário, através do FA e FE, pode desfazer ou refazer algo que envolva persistência de dados no sistema. Por exemplo: pode excluir ou alterar dados inseridos.
<b>1BE. Prevenção de Erros</b>	
<b>1BE1</b>	Verifique se os dados ou opções obrigatórias estão claramente definidos;
<b>1BG. Flexibilidade e eficiência de uso</b>	
<b>1BG1</b>	Verifique se há como passos serem realizados de forma mais rápida e eficiente para usuários mais experientes no FP, FA, FE e RN <sup>22</sup> ;
<b>1BG2</b>	Verifique se o sistema permite que o usuário realize o que ele deseja com facilidade através dos passos do FA e FE;
<b>1BG3</b>	Verifique se o sistema permite ao usuário formas de acesso às tarefas principais no FA e FE. Por exemplo: estando no FA ou FE o usuário consegue voltar ao FP quando for viável;
<b>1BG4</b>	Verifique se as formas de acesso oferecidas pelo sistema no FA e FE diminuem o esforço de ações físicas dos usuários.

---

<sup>19</sup> FP: Fluxo Principal

<sup>20</sup> FA: Fluxo Alternativo

<sup>21</sup> FE: Fluxo Exceção

<sup>22</sup> RN: Regras de Negócio

<b>1BI. Proporcionar funções que são úteis para o usuário</b>	
<b>1BI1</b>	Verifique se tudo que é essencial para o usuário está presente no FP, FA, FE e RN;
<b>1BI2</b>	Verifique se tudo que está presente no FP, FA, FE e RN é útil.

<b>MIT 2 – Avaliação de Usabilidade dos <i>Mockups</i></b>	
<b>Objetivo:</b>	Verificar a usabilidade dos <i>Mockups</i>
<b>Instruções:</b>	<p>1. Antes de começar a inspeção, observe as seguintes diretrizes:</p> <ol style="list-style-type: none"> <li>Caso o <i>Mockup</i> apresente as mensagens do sistema (mensagens de erro, textos informativos, advertências) comece a inspeção utilizando as heurísticas da <b>MIT2 - Alto Detalhamento</b>.</li> <li>Caso o <i>Mockup</i> <b>NÃO</b> apresente as mensagens do sistema, comece a inspeção utilizando as heurísticas da <b>MIT2 - Baixo Detalhamento</b>.</li> </ol> <p>2. Para cada <i>Mockups</i> a ser inspecionado observe e anote qual problema de usabilidade você encontrou seguindo as heurísticas a seguir:</p>
<b>MIT2 - Alto Detalhamento</b>	
<b>2AA. Visibilidade do Status do Sistema</b>	
<b>2AA1</b>	Verifique se há informações textuais ou nome nos <i>Mockups</i> que informa em que parte do sistema o usuário se encontra.
<b>2AA2</b>	Verifique se há algum texto informativo ou mensagem que informa ao usuário o que foi realizado após uma persistência de dados (alteração, exclusão, etc).
<b>2AB. Concordância entre o sistema e o mundo real</b>	
<b>2AB1</b>	Verifique se os símbolos, palavras, frases, nome de botões, links, mensagens de erros e textos informativos estão expressas em uma representação facilmente compreendida pelo usuário, ou seja, possuem uma linguagem familiar;
<b>2AB2</b>	Verifique se as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>2AC. Controle e liberdade ao usuário</b>	
<b>2AC1</b>	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;
<b>2AC2</b>	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado.
<b>2AD. Consistência e padrões</b>	
<b>2AD1</b>	Verifique se a terminologia, gráficos e símbolos estão consistentes;
<b>2AD2</b>	Verifique se há padrões adotados em relação a layout, formatação e controles. Por exemplo, botão “Salvar” no canto superior direito em um <i>Mockup</i> e em outro no canto inferior direito;

<b>2AD3</b>	Verifique se todos os botões/links possuem nomes/ícones que indicam funcionalidades diferentes, ou seja, não há botões/links com nomes/ícones diferentes, mas com a mesma funcionalidade;
<b>2AD4</b>	Verifique se as mensagens de erro são apresentadas de forma consistente com os padrões.
<b>2AE. Prevenção de Erros</b>	
<b>2AE1</b>	Verifique se os dados obrigatórios na entrada de dados estão claramente definidos;
<b>2AE2</b>	Verifique se há alguma indicação do formato correto para uma entrada de dados específica;
<b>2AE3</b>	Verifique se há alguma advertência do sistema que alerta o usuário através de mensagens ou textos informativos de que o que ele está fazendo pode ser inapropriado naquele momento;
<b>2AE4</b>	Verifique se todas as opções, botões e links disponíveis possuem nomes que definem claramente que resultados ou estados serão atingidos.
<b>2AF. Reconhecer ao invés de lembrar</b>	
<b>2AF1</b>	Verifique se os nomes das opções, campos, botões e links são informados de forma que o usuário não tenha que se lembrar quais são eles;
<b>2AF2</b>	Verifique se os ícones/figuras minimizam ou reduzem a carga cognitiva e o esforço físico do usuário de lembrar;
<b>2AF3</b>	Verifique se é fácil reconhecer/visualizar a opção que deve ser usada para atingir o objetivo desejado;
<b>2AG. Flexibilidade e eficiência de uso</b>	
<b>2AG1</b>	Verifique se a disposição dos elementos aumenta a eficiência de uso, minimizando o esforço de ações físicas;
<b>2AG2</b>	Verifique se há apoio a tarefas específicas frequentemente repetidas;
<b>2AG3</b>	Verifique se há diferentes formas de acesso às tarefas principais, ou seja, possibilidade de utilização de aceleradores ou atalhos;
<b>2AG4</b>	Verifique se há facilidade para entrada de dados, sejam eles simples ou complexos estruturalmente;
<b>2AG5</b>	Verifique se há mecanismos de busca de informação que auxiliem a entrada de dados obrigatória;
<b>2AG6</b>	Verifique se há nomes de links abreviados para usuários mais experientes;
<b>2AG7</b>	Verifique se há como as funcionalidades do sistema serem realizadas de forma mais rápida e eficiente para usuários mais experientes.
<b>2AH. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>2AH1</b>	Verifique se as mensagens de erros estão claramente visíveis para o usuário;
<b>2AH2</b>	Verifique se as mensagens de erros utilizam linguagem simples de ser entendida pelos usuários, ou seja, de acordo com o perfil do usuário.
<b>2AH3</b>	Verifique se todas as mensagens de erros não culpam ou intimidam o usuário;

2AH4	Verifique se as mensagens de erro ajudam o usuário a corrigir o erro.
2AH5	Verifique se o sistema mostra como acessar as soluções alternativas que são apresentadas em algumas nas mensagens de erro.
<b>MIT-2AI. Proporcionar funções que são úteis para o usuário</b>	
2AI1	Verifique se toda funcionalidade do sistema, botão ou link essencial está presente;
2AI2	Verifique se toda funcionalidade do sistema, botão ou link presente é útil.
<b>MIT2 - Baixo Detalhamento</b>	
<b>2BA. Visibilidade do Status do Sistema</b>	
2BA1	Verifique se há informações textuais ou nome nos <i>Mockups</i> que informa em que parte do sistema o usuário se encontra.
2BA2	Verifique se há algum texto que informa ao usuário o que foi realizado após uma persistência de dados (alteração, exclusão, etc).
<b>2BB. Concordância entre o sistema e o mundo real</b>	
2BB1	Verifique se os símbolos, palavras, frases, nome de botões, links e textos informativos estão expressas em uma representação facilmente compreendida pelo usuário, ou seja, possuem uma linguagem familiar;
2BB2	Verifique se as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.
<b>2BC. Controle e liberdade ao usuário</b>	
2BC1	Verifique se o usuário tem as opções de desfazer ou refazer algo que ele tenha escolhido;
2BC2	Verifique se o usuário tem a opção de cancelar o que está realizando, ou se há opções similares que permitam ao usuário utilizar saídas em caso de escolhas erradas ou para sair de um estado ou local não esperado.
<b>2BD. Consistência e padrões</b>	
2BD1	Verifique se a terminologia, gráficos e símbolos estão consistentes;
2BD2	Verifique se há padrões adotados em relação a layout, formatação e controles. Por exemplo, botão “Salvar” no canto superior direito em um <i>Mockup</i> e em outro no canto inferior direito;
2BD3	Verifique se todos os botões/links possuem nomes/ícones que indicam funcionalidades diferentes, ou seja, não há botões/links com nomes/ícones diferentes, mas com a mesma funcionalidade;
<b>2BE. Prevenção de Erros</b>	
2BE1	Verifique se os dados obrigatórios na entrada de dados estão claramente definidos;
2BE2	Verifique se há alguma indicação do formato correto para uma entrada de dados específica;
2BE3	Verifique se todas as opções, botões e links disponíveis possuem nomes que definem claramente que resultados ou estados serão atingidos.

<b>2BF. Reconhecer ao invés de lembrar</b>	
<b>2BF1</b>	Verifique se os nomes das opções, campos, botões e links são informados de forma que o usuário não tenha que se lembrar quais são eles;
<b>2BF2</b>	Verifique se os ícones/figuras minimizam ou reduzem a carga cognitiva e o esforço físico do usuário de lembrar;
<b>2BF3</b>	Verifique se é fácil reconhecer/visualizar a opção que deve ser usada para atingir o objetivo desejado;
<b>2BG. Flexibilidade e eficiência de uso</b>	
<b>2BG1</b>	Verifique se a disposição dos elementos aumenta a eficiência de uso, minimizando o esforço de ações físicas;
<b>2BG2</b>	Verifique se há apoio a tarefas específicas frequentemente repetidas;
<b>2BG3</b>	Verifique se há diferentes formas de acesso às tarefas principais, ou seja, possibilidade de utilização de aceleradores ou atalhos;
<b>2BG4</b>	Verifique se há facilidade para entrada de dados, sejam eles simples ou complexos estruturalmente;
<b>2BG5</b>	Verifique se há mecanismos de busca de informação que auxiliem a entrada de dados obrigatória;
<b>2BG6</b>	Verifique se há nomes de links abreviados para usuários mais experientes;
<b>2BG7</b>	Verifique se há como as funcionalidades do sistema serem realizadas de forma mais rápida e eficiente para usuários mais experientes.
<b>MIT-2BI. Proporcionar funções que são úteis para o usuário</b>	
<b>2BI1</b>	Verifique se toda funcionalidade do sistema, botão ou link essencial está presente;
<b>2BI2</b>	Verifique se toda funcionalidade do sistema, botão ou link presente é útil.

<b>MIT 3 – Avaliação de Usabilidade dos Diagramas de Atividades</b>	
<b>Objetivo:</b>	Verificar a usabilidade dos Diagramas de Atividades
<b>Instruções:</b>	1. Para cada Diagrama de Atividades a ser inspecionado observe e anote qual problema de usabilidade você encontrou seguindo as heurísticas baixas:
<b>3B. Concordância entre o sistema e o mundo real</b>	
<b>3B1</b>	Verifique se as atividades estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema;
<b>3B2</b>	Verifique se os nomes das atividades utilizam termos (palavras) que seguem as convenções do mundo real, ou seja, que englobam tanto as convenções do domínio do problema quanto às convenções de terminologia de aplicações semelhantes.
<b>MIT-3C. Controle e liberdade ao usuário</b>	
<b>3C1</b>	Verifique se o usuário através de alguma atividade consegue alterar dados que ele

	inseriu através de outra atividade;
<b>3C2</b>	Verifique se o usuário através de alguma atividade consegue excluir dados que ele inseriu através de outra atividade.
<b>3C3</b>	Verifique se o usuário através de alguma atividade consegue cancelar, voltar ou desfazer algo.
<b>MIT-3D. Consistência e padrões</b>	
<b>3D1</b>	Verifique se todas as atividades indicam funcionalidades diferentes, ou seja, não há atividades com nomes diferentes, mas com a mesma funcionalidade.
<b>MIT-3E. Prevenção de Erros</b>	
<b>3E1</b>	Verifique se há alguma atividade em que o sistema previne algum tipo de erro.
<b>MIT-3F. Reconhecer ao invés de lembrar</b>	
<b>3F1</b>	Verifique se é fácil reconhecer a atividade que deve ser usada para atingir o objetivo desejado.
<b>MIT-3G. Flexibilidade e eficiência de uso</b>	
<b>3G1</b>	Verifique se o usuário possui diferentes formas de acesso às atividades principais;
<b>3G2</b>	Verifique se o usuário pode navegar com facilidade pelas diferentes atividades, ou seja, as possibilidades de navegação para atingir um objetivo estão claras para o usuário.
<b>MIT-3H. Reconhecimento, diagnóstico e recuperação de erros</b>	
<b>3H1</b>	Verifique através das atividades se o sistema ajuda o usuário a sair de uma situação de erro;
<b>3H2</b>	Verifique através das atividades se o sistema ajuda a corrigir o erro. Por exemplo: atividades que indicam a recuperação do erro.
<b>MIT-3I. Proporcionar funções que são úteis para o usuário</b>	
<b>3I1</b>	Verifique se todas as atividades são úteis;
<b>3I2</b>	Verifique se todas as atividades úteis estão presentes.

## APÊNDICE C - Instruções

### Instruções para Avaliação com a AH

1. Anote o horário inicial da inspeção na Planilha de Discrepância.
2. Leia e/ou analise com atenção todo o Modelo (Caso de Uso, *Mockup* ou Diagrama de Atividades).
3. Leia com atenção a heurísticas da Avaliação Heurística.
4. Procure por problemas de usabilidade usando as heurísticas da Avaliação Heurística e anote na Planilha de Discrepância.
5. Anote o horário final da inspeção na Planilha de Discrepância.
6. Calcule o tempo de inspeção (Horário Final – Horário Inicial)

### Instruções para Avaliação com as MITs

1. Anote o horário inicial da inspeção na Planilha de Discrepância.
2. Leia e/ou analise com atenção todo o Modelo (o modelo avaliado pela MIT 1 é o Caso de Uso, na MIT 2 é o *Mockup* e na MIT 3 é o Diagrama de Atividades).
3. Leia com atenção as heurísticas da Técnica MIT.
4. Procure por problemas de usabilidade usando as heurísticas da Técnica MIT e anote na Planilha de Discrepância.
5. Anote o horário final da inspeção na Planilha de Discrepância.
6. Calcule o tempo de inspeção (Horário Final – Horário Inicial)

## APÊNDICE D – Planilhas para anotações das discrepâncias

### Planilhas de discrepâncias para Avaliação com as MITs

#### Planilha de Discrepâncias nos Casos de Uso usando a MIT

Inspetor:

Tempo de inspeção:  minutos

Nro	Item de Verificação	Nro do Caso de Uso	Passo do Caso de Uso	Descrição do Problema Encontrado
1				
2				
3				

#### Planilha de Discrepâncias nos Mockups - usando a MIT

Inspetor:

Tempo de inspeção:  minutos

Nro	Item de Verificação	Nro do Mockup	Descrição do Problema Encontrado
1			
2			
3			

#### Planilha de Discrepâncias nos Diagramas de Atividades - usando a MIT

Inspetor:

Tempo de inspeção:  minutos

Nro	Item de Verificação	Nro do Diagrama de Atividade	Descrição do Problema Encontrado
1			
2			
3			

## Planilhas de discrepâncias para Avaliação com a AH

### Planilha de Discrepâncias nos Casos de Uso usando a AH

Inspetor: \_\_\_\_\_

Tempo de inspeção: \_\_\_\_\_ minutos

Nro	Heurística	Nro do Caso de Uso	Passo do Caso de Uso	Descrição do Problema Encontrado
1				
2				
3				

### Planilha de Discrepâncias nos Mockups - usando a AH

Inspetor: \_\_\_\_\_

Tempo de inspeção: \_\_\_\_\_ minutos

Nro	Heurística	Nro do Mockup	Descrição do Problema Encontrado
1			
2			
3			

### Planilha de Discrepâncias nos Diagramas de Atividades - usando a AH

Inspetor: \_\_\_\_\_

Tempo de inspeção: \_\_\_\_\_ minutos

Nro	Heurística	Nro do Diagrama de Atividade	Descrição do Problema Encontrado
1			
2			
3			