



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



WEB DUE: UMA TÉCNICA DE INSPEÇÃO DE USABILIDADE DE
MOCKUPS DE APLICAÇÕES WEB GUIADA POR ZONAS PRÓPRIAS
DE PÁGINAS WEB

Luis Jorge Enrique Rivero Cabrejos

Dissertação de Mestrado apresentada ao
Programa de Pós-graduação em
Informática, PPGI, da Universidade Federal
do Amazonas, como parte dos requisitos
necessários para a obtenção do título de
Mestre em Informática.

Orientadora: Tayana Uchôa Conte

Manaus
Fevereiro de 2013

WEB DUE: UMA TÉCNICA DE INSPEÇÃO DE USABILIDADE DE
MOCKUPS DE APLICAÇÕES WEB GUIADA POR ZONAS PRÓPRIAS
DE PÁGINAS WEB

Luis Jorge Enrique Rivero Cabrejos

Dissertação de mestrado apresentada ao corpo docente do Instituto de Computação do Programa de Pós-Graduação em Informática (PPGI) da Universidade Federal do Amazonas como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática.

Defesa em 20 de Fevereiro de 2013, em Manaus, Amazonas, pela banca examinadora constituída pelos seguintes professores:

Profª. Tayana Uchôa Conte, D.Sc.

Prof. Raimundo da Silva Barreto, D.Sc

Prof. Gleison dos Santos Souza, D.Sc

Manaus

Fevereiro de 2013

Aos Meus Pais Luis e Gina.

Agradecimentos

Aos meus pais, Gina e Luis, por ter me apoiado desde o início. Embora eles não entendessem muito bem o que significava cada trabalho aceito, sempre ficaram felizes e me incentivaram a correr atrás de mais.

Ao Davi Viana, pelo seu apoio constante, companheirismo, e pelo tempo que passamos durante essa jornada, tanto dentro e fora da faculdade. Muito bom ter você do meu lado e te considerar minha família no Brasil.

À minha amiga Priscila Fernandes, que me apresentou a pesquisa em usabilidade e engenharia de software. Sem ela, hoje não teria atingido essas metas nem conquistado meu mestrado. Muito obrigado.

Ao Bruno Bonifácio, pelo companheirismo e pelo ombro amigo, com ele desabafei muito e consegui sentir paz em momentos em que mais precisava de alguém para me ouvir.

Ao Marcos Gomes, quem sempre me lembrava de respirar um pouco no meio de tanto trabalho. Obrigado pelos descansos e as boas conversas.

À Claudia Castillo e à Veronica Moreno, minhas amigas do Perú. A gente continua se vendo, com pouca frequência, mas como se nada tivesse acontecido. É bom voltar para casa e ver que pouca coisa mudou.

Aos meus colegas do USES, Olavo, Bia, Jacy, Natasha, Amadeu, Fortaleza e Serginho. Obrigado pelo tempo que passamos juntos, pelas críticas construtivas, pelas experiências e por formar um grupo de pesquisa que parecia mais uma família onde todos trabalhavam para crescer e melhorar.

Ao grupo do sistema SiON, Andrea Waichman, Horácio Fernandes, David Fernandes, com os que aprendi muito sobre o dia a dia da indústria de desenvolvimento de software. Foi ótimo trabalhar com pessoas tão qualificadas e aprender cada dia mais.

Ao Raimundo Barreto e ao Gleison Santos por aceitarem participar na minha banca de mestrado. Com certeza qualquer crítica ou sugestão que tiverem ajudarão a amadurecer e aumentar a qualidade do meu trabalho.

A todos os participantes dos estudos experimentais pela sua colaboração.

À Universidade Federal do Amazonas e ao Instituto de Computação, pelo seu apoio durante todo o mestrado. Em especial à Elienai Nogueira e a Helen Nascimento pela ajuda em todos os momentos.

À CAPES pelo apoio financeiro ao longo do mestrado.

Finalmente, um agradecimento especial à minha orientadora, a professora Tayana Conte. Professora, com a senhora aprendi muito: a trabalhar em equipe, a gerenciar meu tempo, a aplicar meu conhecimento na prática, a buscar respostas, a solucionar os problemas que vierem, a ser proativo, a coordenar. Aprendi a fazer pesquisa e é pela senhora que cresci e atingi as metas que me propus. Muito obrigado mesmo por ser uma orientadora em todos os sentidos.

Índice

1	Introdução.....	1
1.1	Contexto	1
1.2	Problema	2
1.3	Objetivos	3
1.4	Metodologia.....	4
1.5	Organização.....	8
2	Inspeções de Usabilidade de Aplicações <i>Web</i>	10
2.1	Introdução.....	10
2.2	Inspeções de Usabilidade	11
2.2.1	Avaliação de Usabilidade.....	11
2.2.2	Métodos de Inspeção de Usabilidade.....	12
2.2.3	Motivação para uma revisão sobre <i>UIMs</i>	15
2.3	Procedimento de Execução da Extensão do Mapeamento Sistemático	15
2.3.1	Seleção de Estudos Primários.....	16
2.3.2	Estratégia de Extração de Dados	18
2.3.3	Execução	19
2.4	Resultados	21
2.4.1	Método Teórico Base.....	22
2.4.2	Tipo de Artefato Avaliado	23
2.4.3	Tipo de Aplicação <i>Web</i> Avaliada pelo método de Inspeção	24
2.4.4	Uso de Inspetores no processo de inspeção	25
2.5	Resumo dos Métodos de Inspeção de Usabilidade de Aplicações <i>Web</i>	25
2.6	Discussão dos Resultados da Extensão do Mapeamento Sistemático	30
2.6.1	Principais Descobertas	30
2.6.2	Implicações para a Academia e a Indústria.....	31
2.7	Considerações Finais	33
3	A Técnica <i>Web Design Usability Evaluation</i>	35
3.1	Introdução.....	35
3.2	Apoiando a Indústria de Desenvolvimento de Software.....	36
3.3	A Concepção da Técnica <i>Web Due</i>	38

3.3.1	Avaliação da Usabilidade de Protótipos de Baixa Fidelidade (ou Mockups).....	38
3.3.2	Processo de Inspeção Guiado por Zonas Próprias de Páginas <i>Web</i>	38
3.3.3	Uso de Apoio Ferramental para reduzir o Esforço dos Inspetores.....	42
3.4	Aplicando a Técnica <i>Web DUE</i> na Inspeção de <i>Mockups</i>	43
3.4.1	O Processo de Inspeção da Técnica <i>Web DUE</i>	43
3.4.2	Prova de Conceito da Técnica <i>Web DUE</i>	44
3.5	Considerações Finais	49
4	O Estudo de Viabilidade da Técnica <i>Web DUE</i>	51
4.1	Introdução.....	51
4.2	Estudo de Viabilidade	52
4.2.1	Objetivo.....	52
4.2.2	Hipóteses	53
4.2.3	Participantes	53
4.2.4	Materiais.....	55
4.2.5	Procedimento e Coleta de Dados.....	56
4.3	Análise dos Resultados do Estudo de Viabilidade	57
4.3.1	Análise Quantitativa	58
4.3.2	Análise Qualitativa	60
4.4	Análise dos Resultados do Estudo de Viabilidade	64
4.4.1	Validade Interna	64
4.4.2	Validade Externa	65
4.4.3	Validade de Conclusão	65
4.4.4	Validade de Constructo.....	66
4.5	Considerações Finais	66
5	A Ferramenta <i>Mockup Design Usability Evaluation</i>	68
5.1	Introdução.....	68
5.2	A Concepção da Ferramenta <i>Mockup DUE</i>	69
5.3	Aplicando a Primeira Versão da Ferramenta <i>Mockup DUE</i>	70
5.3.1	Planejamento da Inspeção	71
5.3.2	Detecção de Problemas e Relatório	74
5.4	Considerações Finais	79
6	O Estudo de Viabilidade da Ferramenta <i>Mockup DUE</i>	81
6.1	Introdução.....	81

6.2	Estudo de Viabilidade	82
6.2.1	Objetivo.....	82
6.2.2	Participantes	82
6.2.3	Materiais.....	83
6.2.4	Procedimento.....	84
6.2.5	Coleção de Dados	85
6.3	Análise Qualitativa	85
6.4	Ameaças à Validade.....	90
6.5	Considerações Finais	91
7	A Evolução da Técnica <i>Web DUE</i> e da Ferramenta <i>Mockup DUE</i>	92
7.1	Introdução.....	92
7.2	A Técnica <i>Web DUE</i> v2.....	93
7.3	A Ferramenta <i>Mockup DUE</i> v2.....	98
7.4	Proposta da nova versão da Ferramenta <i>Mockup Design Usability Evaluation</i> 101	
7.5	Considerações Finais	103
8	Conclusões e Perspectivas Futuras	105
8.1	Epílogo	105
8.2	Contribuições.....	106
8.3	Perspectivas Futuras.....	108
8.3.1	Desenvolvimento e Avaliação da Ferramenta <i>Mockup G-DUE</i>	108
8.3.2	Avaliações Adicionais	108
8.3.3	Transferência das Tecnologias para a Indústria	109
	Referências	110
	Apêndice A: Referências da Extensão do Mapeamento Sistemático	114
	Apêndice B: A Técnica <i>Web DUE</i> v1.....	117
	Instruções para Identificação de Problemas Usando a Técnica <i>Web DUE</i>	117
	Fluxograma de Atividades do Processo de Detecção da Técnica <i>Web DUE</i> v1.....	118
	Itens de Verificação da Zona de Navegação.....	119
	Itens de Verificação da Zona de Estado do Sistema	121
	Itens de Verificação da Zona de Ajuda	123
	Itens de Verificação da Zona de Informação	125
	Itens de Verificação da Zona de Serviços.....	127
	Itens de Verificação da Zona de Informação do Usuário	129

Itens de Verificação da Zona de Acesso Direto.....	131
Itens de Verificação da Zona de Entrada de Dados	133
Itens de Verificação da Zona de Instituição.....	137
Itens de Verificação da Zona de Customização	138
Apêndice C: A Técnica <i>Web DUE</i> v2.....	140
Instruções para Identificação de Problemas Usando a Técnica <i>Web DUE</i>	140
Fluxograma de Atividades do Processo de Detecção da Técnica <i>Web DUE</i> v2.....	142
Itens de Verificação da Zona de Navegação.....	143
Itens de Verificação da Zona de Estado do Sistema	145
Itens de Verificação da Zona de Ajuda	147
Itens de Verificação da Zona de Informação	149
Itens de Verificação da Zona de Serviços.....	151
Itens de Verificação da Zona de Informação do Usuário	153
Itens de Verificação da Zona de Acesso Direto.....	155
Itens de Verificação da Zona de Entrada de Dados	157
Itens de Verificação da Zona de Instituição.....	161
Itens de Verificação da Zona de Customização	162

Índice de Figuras

Figura 1-1 Metodologia adotada para a definição da técnica <i>Web DUE</i> de inspeção de usabilidade de mockups de aplicações <i>Web</i>	7
Figura 2-1 Etapas do processo de execução da extensão do mapeamento sistemático com base no mapeamento em [12]. Figura adaptada de [29].	16
Figura 3-1 Extrato da técnica WDP [6] contendo parte dos itens de verificação.	41
Figura 3-2 Processo de inspeção da técnica <i>Web DUE</i>	44
Figura 3-3 <i>Mockup</i> #1 do portal do MPS.Br: página de listagem de guias.	45
Figura 3-4 <i>Mockup</i> #2 do portal do MPS.Br: página apresentando um guia.	46
Figura 3-5 Divisão dos <i>mockups</i> em zonas próprias de páginas <i>Web</i>	47
Figura 3-6 Elementos da interface associados aos itens de verificação violados.	49
Figura 4-1 Imagens dos <i>mockups</i> usados no processo de inspeção e um dos mapeamentos para simular a interação.	56
Figura 4-2 Processo de inspeção no estudo de viabilidade.	57
Figura 4-3 Boxplot com o comparativo de eficácia.	59
Figura 4-4 Boxplot com o comparativo de eficiência.	60
Figura 4-5 Questionário pós-avaliação da técnica <i>Web DUE</i>	61
Figura 5-1 Tela inicial da ferramenta <i>Mockup DUE</i>	71
Figura 5-2 Iniciando um planejamento de inspeção.	72
Figura 5-3 Tela de planejamento de inspeção.	73
Figura 5-4 Funcionalidades do planejamento de inspeção.	74
Figura 5-5 Iniciando o processo de detecção de problemas de usabilidade.	75
Figura 5-6 Tela de detecção de problemas de usabilidade.	77
Figura 5-7 Funcionalidades do processo de detecção de problemas de usabilidade.	78
Figura 5-8 Exemplo de relatório gerado com a ferramenta <i>Mockup DUE</i>	79
Figura 6-1 Participante aplicando a ferramenta <i>Mockup DUE</i>	85
Figura 6-2 Resultados gerais de satisfação dos participantes.	86

Figura 6-3 Questões relacionadas à facilidade de uso da ferramenta.	86
Figura 7-1 Exemplo onde os itens de verificação da técnica Web DUE podem e não ser avaliados.	96
Figura 7-2 Exemplo de item renomeado para evitar a identificação de falsos positivos.	97
Figura 7-3 Exemplo de exemplo/esclarecimento reescrito na técnica <i>Web DUE</i> v2.	97
Figura 7-4 Informando o estado do sistema na ferramenta <i>Mockup DUE</i> v2.	98
Figura 7-5 Renomeando botões para facilitar o entendimento na <i>Mockup DUE</i> v2.	99
Figura 7-6 Solicitando confirmação na ferramenta <i>Mockup DUE</i> v2.	99
Figura 7-7 Modificações na atividade de detecção de defeitos.	100
Figura 7-8 Localização das mudanças na proposta da <i>Mockup G-DUE</i>	101
Figura 7-9 Identificando zonas e verificando itens na proposta da <i>Mockup G-DUE</i>	102
Figura 7-10 Identificando de problemas de usabilidade na <i>Mockup G-DUE</i>	103

Índice de Tabelas

Tabela 1-1 Perguntas respondidas na metodologia experimental de Shull <i>et al.</i> [35].....	5
Tabela 2-1 Lista das heurísticas propostas por Nielsen [26].....	13
Tabela 2-2 Sub-questões de pesquisa do mapeamento sistemático de Fernandez <i>et al.</i> [12] e suas possíveis respostas.....	17
Tabela 2-3 Sub-questões de pesquisa para a execução da extensão do mapeamento sistemático e sua motivação.	18
Tabela 2-4 Código, título, autores e veículo de publicação dos artigos analisados na extensão do mapeamento sistemático.	20
Tabela 2-5 Resultados da classificação de estudos primários sobre UIMs para a <i>Web</i> ..	22
Tabela 3-1 Resumo dos resultados do mapeamento sistemático apresentados no Capítulo 2.	37
Tabela 3-2 Lista de zonas de páginas <i>Web</i> baseadas em [13], conteúdo e adequação para sua inclusão em uma página <i>Web</i>	39
Tabela 3-3 Descrição das perspectivas da técnica WDP [6] e seus respectivos relacionamentos com as heurísticas de Nielsen [26].	40
Tabela 3-4 Parte do <i>checklist</i> de itens de verificação da zona de entrada de dados.....	42
Tabela 3-5 Itens de verificação violados pelos <i>mockups</i> do portal do MPS.Br.	48
Tabela 4-1 Objetivo do estudo de viabilidade da técnica <i>Web DUE</i>	53
Tabela 4-2 Resultados do estudo experimental por participante e técnica.	58
Tabela 6-1 Resultados da classificação dos participantes da avaliação cooperativa.....	83
Tabela 6-2 Mudanças sugeridas pelos inspetores durante a avaliação cooperativa.	89
Tabela 7-1 Itens de verificação que levaram os inspetores a apontar falsos positivos...	94

Resumo da Dissertação apresentada ao PPGI/UFAM como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática.

WEB DUE: UMA TÉCNICA DE INSPEÇÃO DE USABILIDADE DE MOCKUPS DE APLICAÇÕES WEB GUIADA POR ZONAS PRÓPRIAS DE PÁGINAS WEB

Luis Jorge Enrique Rivero Cabrejos

Fevereiro/2013

Orientadora: Tayana Uchôa Conte

Programa: Programa de Pós Graduação em Informática

Apesar do aumento do número de métodos de inspeção de usabilidade de aplicações Web, a maioria destes métodos só é aplicada nas últimas etapas do processo de desenvolvimento, aumentando o custo de correção dos problemas encontrados. Esta dissertação de mestrado apresenta a técnica *Web Design Usability Evaluation (Web DUE)*, uma técnica específica para inspeções de usabilidade de protótipos em papel (ou *mockups*). A técnica *Web DUE* tem por finalidade apoiar os inspetores na identificação de problemas de usabilidade durante as primeiras etapas do processo de desenvolvimento, visando assim à redução dos custos de correção dos mesmos. Para atingir esse objetivo, a técnica *Web DUE* sugere a divisão dos *mockups* em zonas de páginas Web e a avaliação de usabilidade dos mesmos com base em um conjunto de itens de verificação. Para simular a interação entre os *mockups* e o usuário e apoiar a detecção de defeitos de usabilidade, foi criada a ferramenta *Mockup DUE*. Tanto a técnica *Web DUE* quanto a ferramenta *Mockup DUE* tiveram sua construção e avaliação apoiada por experimentação. Esta dissertação apresenta: (a) como a técnica *Web DUE* foi elaborada com base na análise da extensão de um mapeamento sistemático; (b) o estudo de viabilidade da técnica *Web DUE*; (c) a criação da ferramenta *Mockup DUE* com base nos resultados do primeiro estudo de viabilidade da técnica; (d) o estudo de viabilidade da ferramenta *Mockup DUE*; e (e) a evolução das tecnologias propostas para a inspeção de *mockups* de aplicações Web.

Palavras-chave: *Mockups*, Inspeção de Usabilidade, Qualidade de *Software*, Estudo Experimental, Apoio Ferramental.

Abstract of the Thesis presented to PPGI/UFAM as a partial fulfillment of the requirements for the degree of Master in Informatics.

WEB DUE: A USABILITY INSPECTION TECHNIQUE FOR MOCKUPS OF WEB APPLICATIONS THAT GUIDES INSPECTORS THROUGH WEB PAGE ZONES

Luis Jorge Enrique Rivero Cabrejos

February/2013

Advisor: Tayana Uchôa Conte

Program: Programa de Pós Graduação em Informática

Despite the increasing number of usability inspection methods for Web applications, most of these methods are being applied in later stages of the development process, which increases the cost of correcting the identified problems. This master degree thesis presents the Web Design Usability Evaluation (Web DUE) technique, a technique that was specifically crafted for the usability inspection of paper based prototypes (or mockups). The Web DUE technique aims at aiding inspectors in the identification of usability problems in earlier stages of the development process, thus reducing the cost of correcting them. To achieve this goal, the Web DUE technique suggests to divide mockups into Web page zones and, using a set of usability verification items, to check if the interface is usable or presents usability problems. To simulate interaction between the mockups and the user, and to support usability problems detection, we created the Mockup DUE tool. Both, the creation of the Web DUE technique and Mockup DUE tool, were supported by empirical validation. In this dissertation we present: (a) how we created the Web DUE technique based on the analysis of the results of a systematic mapping extension; (b) the feasibility study of the technique; (c) how the results from the empirical study helped in the creation of the Mockup DUE tool; (d) the feasibility study of the Mockup DUE tool; and (e) the evolution of the proposed technologies for the inspection of Web mockups.

Keywords: Mockups, Usability Inspection, Software Quality, Empirical Study, Tool Support.

1 Introdução

Este capítulo apresenta a introdução a esta dissertação de mestrado. Além de contextualizar esta pesquisa, é apresentada a motivação, objetivos e a metodologia seguida. O restante deste capítulo apresenta a estrutura desta dissertação de mestrado sobre inspeção de usabilidade de mockups de aplicações Web.

1.1 Contexto

Atualmente as aplicações *Web* são o principal suporte do comércio e da troca de informações, e estão sendo usadas para apresentar produtos e serviços a clientes potenciais [12]. Essa dependência nas aplicações *Web* tem sido refletida no aumentado consideravelmente do seu uso nos últimos anos [25].

Murugesan [25] afirma que o número de aplicações *Web* tem experimentado um crescimento exponencial; e que o número atual de páginas “*deep Web*” (páginas que são montadas em resposta às ações dos usuários) pode chegar a estar entre 400 e 750 bilhões. Segundo o autor, esses números têm despertado o interesse da indústria de desenvolvimento de software para o desenvolvimento de aplicações mais complexas e com um maior número de funcionalidades a fim de produzir produtos mais competitivos no mercado.

O aumento do número de aplicações *Web* também fez com que o nível de expectativa dos usuários aumentasse. Os usuários esperam aplicações *Web* amigáveis que proporcionem uma interface direta e fácil de entender, ou, em outras palavras, aplicações com alto nível de Usabilidade. A ISO 9241-11 [16] define o termo usabilidade como “*a medida pela qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com efetividade, eficiência e satisfação em um contexto de uso específico*”. As aplicações *Web* precisam ser usáveis, pois sua complexidade é alta, lidam com vários tipos de usuário e conteúdo, e possuem diferentes objetivos de comunicação [39].

Na área de Engenharia de Software, a usabilidade é vista como uma contribuição para a qualidade do software [23]. Segundo Matera *et al.* [21], a aceitação

das aplicações *Web* é determinada pelo seu grau de usabilidade, visto que, se uma aplicação *Web* possuir pouca usabilidade, a mesma será rapidamente substituída por outra mais usável assim que seu público alvo souber da sua existência [22].

Apesar do reconhecimento da usabilidade como fator crítico de qualidade, Offut [27] afirma que muitas das aplicações *Web* existentes não são usáveis o suficiente para atingir às expectativas dos usuários. Conseqüentemente, a indústria de desenvolvimento de *software* tem investido no desenvolvimento de um conjunto de métodos de inspeção de usabilidade (*Usability Inspection Methods - UIMs*) para corrigir problemas de usabilidade [21]. *UIMs* são procedimentos com etapas definidas onde especialistas em usabilidade inspecionam ou examinam aspectos relacionados à usabilidade das interfaces dos sistemas de *software* [32].

Diversas pesquisas envolvendo a melhoria da qualidade do *software* por meio de inspeções de usabilidade têm sido publicadas [15]. No entanto, Fernandez *et al.* [12] identificaram que ao redor de 91% dos métodos de avaliação de usabilidade são aplicados na etapa de implementação das aplicações, o que aumenta o custo de correção dos problemas de usabilidade encontrados. Segundo Travassos *et al.* o custo de correção dos problemas encontrados durante o desenvolvimento de *software* diminui se estes forem encontrados no início do desenvolvimento [38].

Diante do exposto, é necessário propor tecnologias com o intuito de garantir a qualidade das aplicações *Web* em termos de usabilidade, durante as primeiras etapas do processo de desenvolvimento. Na seguinte seção será descrito o problema a ser tratado nesta dissertação de mestrado.

1.2 Problema

A inclusão de práticas de usabilidade dentro do processo de *software* guiado pela engenharia de *software* pode contribuir, segundo Donahue *et al.* [8], com: (a) a melhoria das interfaces de usuário; (b) o incremento da produtividade dos usuários; e (c) redução de tempo e custos no desenvolvimento, treinamento, manutenção e documentação de *software*.

Esta pesquisa é motivada pela integração de práticas de melhoria da usabilidade às práticas de engenharia de *software* durante as primeiras etapas do processo de desenvolvimento. Estima-se que o custo da correção dos problemas encontrados após a entrega do *software* chegue a ser 100 vezes maior do que o custo de correção durante as primeiras etapas do processo de desenvolvimento [5].

Nesse contexto, o problema tratado nesta dissertação de mestrado está relacionado com a garantia da qualidade das aplicações *Web* durante as primeiras etapas do processo de desenvolvimento. A questão de pesquisa deste trabalho é:

“Como melhorar o processo de inspeção de usabilidade e conseqüentemente melhorar a usabilidade das aplicações Web a fim de garantir sua qualidade e competitividade na indústria de desenvolvimento de software durante as primeiras etapas do processo de desenvolvimento?”

A questão de pesquisa desse trabalho consiste na possibilidade de definição de uma técnica para inspeção de usabilidade que apoie a obtenção de maior qualidade nas aplicações *Web*, permitindo a identificação de um maior número de problemas de usabilidade durante o *design* da aplicação. Nesse sentido, pretende-se avaliar a usabilidade das aplicações *Web*, inspecionando o *design*, enquanto elas estão sendo projetadas, através da inspeção de protótipos de baixa fidelidade (ou *mockups*). Um método de avaliação de usabilidade do design de aplicações *Web* baseadas em *mockups* pode ajudar a melhorar a usabilidade das interfaces de usuário e detectar problemas antes que a aplicação seja desenvolvida [2].

A hipótese neste trabalho é que a utilização de uma técnica de inspeção que direcione o processo de inspeção por meio de zonas (ou partes específicas) próprias de páginas de aplicações *Web* pode ajudar a melhorar a usabilidade de tais aplicações durante as primeiras etapas do processo de desenvolvimento e permitir: (a) a redução o tempo gasto durante as inspeções, e (b) o aumento do número de problemas de usabilidade identificados. Além disso, espera-se com o uso da técnica, diminuir o custo de correção dos problemas de usabilidade encontrados ao identificá-los nas primeiras etapas do processo de desenvolvimento [38], e contribuir para o aumento da qualidade das aplicações *Web* ao identificar problemas que dificultam a interação dos usuários com o sistema.

1.3 Objetivos

Este trabalho de mestrado tem como objetivo global definir, avaliar experimentalmente e evoluir uma técnica de inspeção de usabilidade de aplicações *Web* que direcione inspetores pelo processo de inspeção durante as etapas de *design* da aplicação *Web*, com base nas zonas das páginas da aplicação *Web* a ser inspecionada. Para alcançar este objetivo geral, buscou-se decompô-lo nos seguintes objetivos específicos:

- Identificar e comparar as diferentes abordagens propostas em inspeções de usabilidade em aplicações *Web* com a finalidade de identificar melhorias e incluí-las na definição da nova técnica.
- Analisar a aplicação das zonas típicas de páginas de aplicações *Web* [13] no processo de inspeção de usabilidade.
- Analisar os itens de verificação da técnica com base nos resultados dos estudos experimentais para propor melhorias e facilitar a sua utilização.
- Avaliar experimentalmente se a técnica atende os atributos de aceitação e adoção necessários no contexto da indústria de desenvolvimento de *software*.
- Propor uma ferramenta que facilite a aplicação da técnica, automatizando parte do seu processo de inspeção e melhorando o desempenho da inspeção. Nesta etapa devem ser incluídas as melhorias sugeridas da análise dos resultados de estudos experimentais.

1.4 Metodologia

Para atingir os objetivos relacionados com esta pesquisa será utilizada uma metodologia baseada em experimentação, que utiliza estudos experimentais para avaliar e evoluir tecnologias de software. A seguir são descritas as metodologias experimentais que serviram de base para compor a metodologia de pesquisa deste trabalho.

A metodologia experimental de Shull *et al.* [35], é uma abordagem que permite evoluir uma tecnologia desde sua definição até a sua transferência para a indústria. Para isso, são feitos estudos experimentais que buscam responder perguntas que examinam os problemas da tecnologia. A Tabela 1-1 mostra as perguntas a serem respondidas, o estudo experimental relacionado com a pergunta, e o propósito do estudo experimental. Segundo Shull *et al.* [35], a ordem das perguntas evita o retrabalho da definição da tecnologia, pois, ela permite:

- Identificar os problemas básicos e fundamentais que podem gerar mudanças drásticas na tecnologia proposta no início da sua definição.
- Identificar os problemas específicos da tecnologia que requerem mudanças menores para serem consertados.

Tabela 1-1 Perguntas respondidas na metodologia experimental de Shull *et al.* [35].

Pergunta	Estudo	Propósito
Os resultados do processo são viáveis e o tempo bem empregado?	Estudo de Viabilidade	Determinar se é possível viabilizar o uso da tecnologia.
Os passos do processo fazem sentido?	Estudo de Observação	Observar a tecnologia a fim de aprimorar o entendimento dos pesquisadores em relação à aplicação da mesma e possibilitar seu refinamento.
O processo é adequado ao ciclo de vida real?	Estudo de Caso (ciclo de vida)	Analisar a tecnologia em um determinado contexto de ciclo de vida e para caracterizar a aplicação da mesma.
O processo é adequado ao ambiente industrial?	Estudo de Caso (indústria)	Identificar se existem problemas de integração na aplicação da tecnologia proposta em um ambiente industrial.

A metodologia experimental de Mafra *et al.* [20] é uma extensão da metodologia de Shull *et al.* [35] para a introdução de tecnologias de software na indústria, que se baseia em estudos experimentais como forma de determinar o que funciona ou não na aplicação da tecnologia proposta. Mafra *et al.* [20] identificaram a necessidade de executar estudos secundários antes dos estudos primários sugeridos em Shull *et al.* [35] para permitir uma melhoria na avaliação das tecnologias emergentes. Os estudos primários são aqueles estudos que o avaliador realiza para gerar novo conhecimento, isto é levantar hipóteses e prová-las. Já os estudos secundários não geram evidências, porém visam identificar, avaliar, e interpretar todos os resultados relevantes a um determinado tópico de pesquisa, fenômeno de interesse ou questão de pesquisa. Nesta visão a metodologia de Mafra *et al.* [20] é dividida em duas partes, sendo que a primeira parte contém a extensão proposta, onde são realizados os estudos secundários, e a segunda contém a metodologia original definida por Shull *et al.* [35].

Na Figura 1-1, é apresentada a metodologia adotada nesta pesquisa. Esta metodologia utiliza parte dos estudos sugeridos em Mafra *et al.* [20]: (a) um estudo secundário para identificar o estado da arte dos métodos de inspeção de usabilidade de aplicações Web, e (b) dois estudos experimentais para a validação da técnica proposta: um estudo de viabilidade da técnica e um estudo de viabilidade da ferramenta. A seguir são detalhadas as atividades a serem realizadas em cada etapa da metodologia.

- Extensão de um Mapeamento Sistemático: Esta etapa consiste na realização ou extensão de um estudo secundário para identificar estudos primários que tratem de métodos de inspeção de usabilidade de aplicações *Web*. Os resultados desta etapa servirão para identificar as necessidades na área de Inspeções de Usabilidade de Aplicações *Web*.
- Proposta Inicial: Esta etapa consiste na criação da primeira versão da técnica *Web DUE*. Para isso, foi feita uma análise de atributos de usabilidade para as diferentes zonas de páginas *Web*. Com base nesse relacionamento, foi proposto um conjunto de *checklists* de verificação de usabilidade que serão usados para a identificação de problemas de usabilidade.
- Execução do Primeiro Estudo de Viabilidade: Esta etapa consiste na execução de um estudo de viabilidade para responder a seguinte questão de pesquisa: *Os resultados da técnica Web DUE são viáveis em termos do número de defeitos encontrados e do tempo gasto para achá-los?* Os resultados deste estudo experimental proporcionam dados que podem ser usados para sugerir melhorias na técnica e assim aprimorar seus indicadores de eficiência e eficácia.
- Levantamento de Requisitos para a Ferramenta: Esta etapa consiste na análise dos resultados obtidos após a execução do primeiro estudo de viabilidade. Com essa análise é levantado um conjunto de requisitos a ser incorporado à proposta do apoio ferramental da técnica *Web DUE*.
- Desenvolvimento do Apoio Ferramental: Esta é a etapa de implementação da Ferramenta *Mockup DUE*. Nesta etapa é automatizado o processo de inspeção da técnica *Web DUE* visando facilitar a interação entre os inspetores e os *mockups*, assim como simplificar a utilização da técnica e a geração de relatórios contendo os problemas de usabilidade.

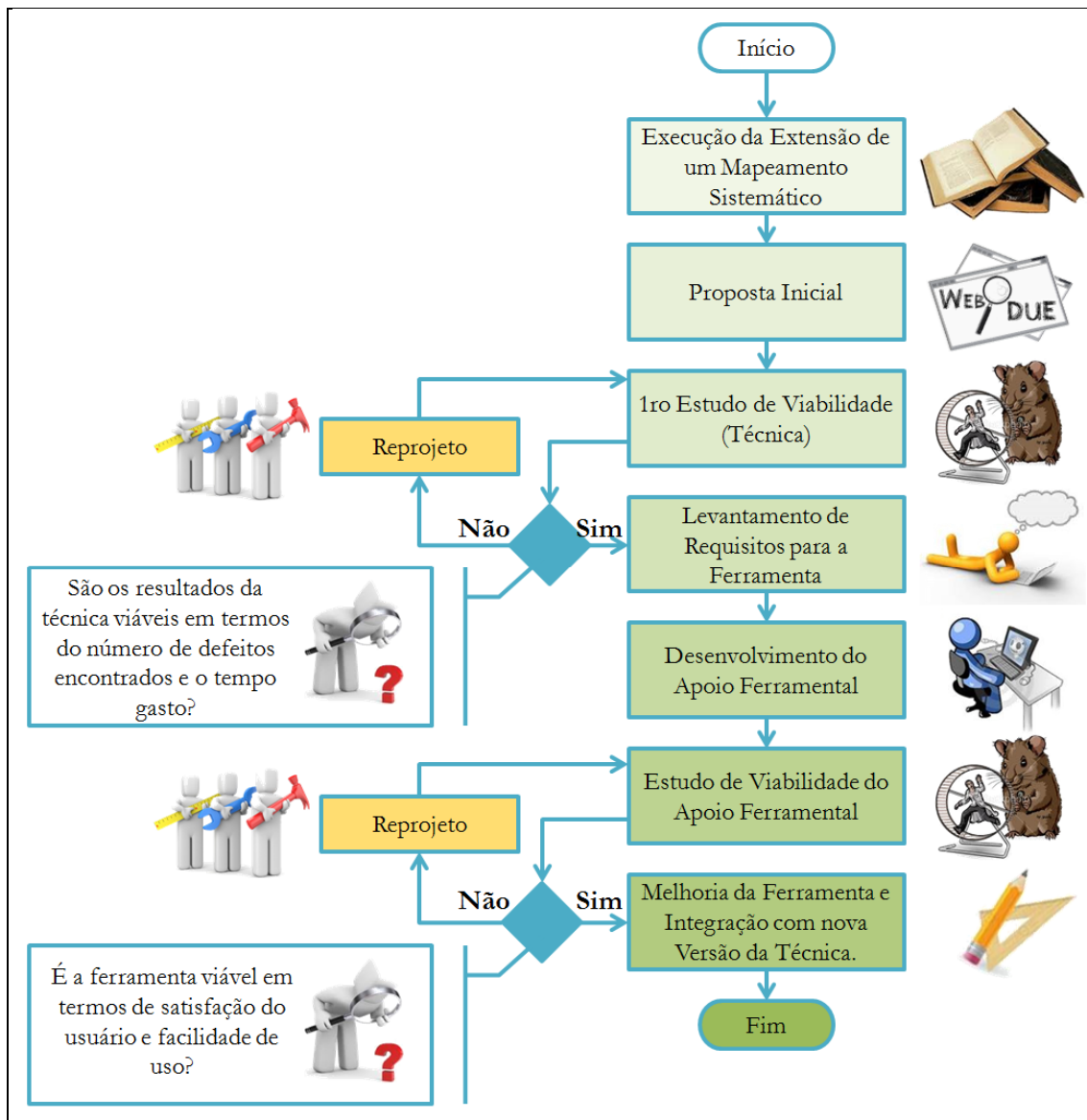


Figura 1-1 Metodologia adotada para a definição da técnica *Web DUE* de inspeção de usabilidade de mockups de aplicações *Web*.

- Estudo de Viabilidade do Apoio Ferramental: Nesta etapa é feito um estudo experimental para responder a seguinte questão de pesquisa: *A ferramenta Mockup DUE é viável em termos de satisfação do usuário e facilidade de uso?* Os resultados deste estudo proporcionam informações sobre o design da Ferramenta *Mockup DUE*, e sugerem melhorias na sua interface e processo de aplicação.
- Melhoria da Ferramenta e Integração com a nova versão da Técnica: Nesta etapa são usados os resultados de ambos os estudos: estudo de viabilidade da técnica e estudo de viabilidade da ferramenta. Com os resultados do estudo de viabilidade da técnica sugerem-se melhorias e é criada a nova versão da mesma. Com os

resultados do estudo de viabilidade ferramenta, visa-se sugerir melhorias na interface da ferramenta. Finalmente, a nova versão da técnica e a nova versão da ferramenta são integradas para melhorar o processo de inspeção de usabilidade de *mockups* de aplicações *Web*.

1.5 Organização

Esta dissertação de mestrado está organizada em outros sete capítulos, além deste primeiro capítulo que apresentou a motivação e o contexto no qual está inserida esta pesquisa. A organização do texto deste trabalho segue a estrutura abaixo:

Capítulo 2 - Inspeções de Usabilidade de Aplicações *Web*: Apresenta a base teórica sobre métodos de inspeção de usabilidade (*Usability Inspection Methods - UIMs*) em aplicações *Web*. São apresentados os resultados de uma extensão de um mapeamento sistemático sobre métodos de avaliação de usabilidade. O objetivo desta extensão foi caracterizar os *UIMs* especificamente propostos para inspecionar a usabilidade de aplicações *Web*.

Capítulo 3 – A Técnica *Web Design Usability Evaluation*: Apresenta a proposta inicial da técnica de inspeção de usabilidade de protótipos em papel (ou *mockups*) de aplicações *Web*. O capítulo descreve a técnica base que foi usada para criar a técnica *Web DUE - Web Design Usability Evaluation*, assim como as zonas próprias de páginas *Web* que foram usadas para direcionar o processo de inspeção. Além disso, é detalhado o processo de inspeção da técnica e um exemplo prático da sua aplicação.

Capítulo 4 – O Estudo de Viabilidade da Técnica *Web DUE*: Descreve o estudo de viabilidade executado com o propósito de avaliar e aperfeiçoar a técnica *Web DUE* em termos de eficácia e eficiência.

Capítulo 5 – A Ferramenta *Mockup Design Usability Evaluation*: Apresenta a proposta inicial da ferramenta de inspeção de usabilidade de protótipos de aplicações *Web*. O capítulo descreve os requisitos que foram integrados na ferramenta *Mockup DUE – Mockup Design Usability Evaluation* com base nos resultados do primeiro estudo de viabilidade da técnica *Web DUE*. Além disso, é detalhado o processo de execução de uma inspeção de usabilidade fazendo uso da ferramenta e um exemplo prático da sua aplicação.

Capítulo 6 – O Estudo de Viabilidade da Ferramenta *Mockup DUE*: Descreve o estudo de viabilidade executado com o propósito de avaliar e aperfeiçoar a ferramenta *Mockup DUE* em termos de satisfação do usuário e facilidade de uso.

Capítulo 7 – A Evolução da Técnica *Web DUE* e da Ferramenta *Mockup DUE*:
Descreve as mudanças realizadas na técnica *Web DUE* e as mudanças no design da
ferramenta *Mockup DUE* com base nos resultados dos estudos de viabilidade.

Capítulo 8 – Conclusão e Perspectivas Futuras: Descreve as conclusões e
contribuições do trabalho, além de indicar a continuação da pesquisa através trabalhos
futuros.

2 Inspeções de Usabilidade de Aplicações Web

Este capítulo apresenta os conceitos relacionados a inspeções de usabilidade em aplicações Web. Além de introduzir os conceitos básicos para contextualizar o uso de inspeções de usabilidade, é apresentada a extensão de um mapeamento sistemático sobre métodos de avaliação de usabilidade. O objetivo desta extensão foi caracterizar os UIMs especificamente propostos para inspecionar a usabilidade de aplicações Web. A análise dos resultados oferece aos pesquisadores e à indústria de software o conhecimento necessário para entender as vantagens e desvantagens dos UIMs emergentes.

2.1 Introdução

Na última década, a indústria de desenvolvimento de software tem investido no desenvolvimento de uma variedade de Métodos de Inspeção de Usabilidade (*Usability Inspection Methods - UIMs*) para corrigir problemas de usabilidade na Web [21]. Segundo Insfran e Fernandez [15], mesmo com o aumento do número de *UIMs* emergentes, as companhias ainda não os estão usando. A não aplicação destes *UIMs* pode ser um dos motivos para a baixa qualidade das aplicações Web.

Um mapeamento sistemático é um tipo de revisão sistemática que permite categorizar e resumir de forma objetiva a informação existente relacionada a uma questão de pesquisa. Fernandez *et al.* [12] executaram um mapeamento sistemático para identificar os Métodos de Avaliação de Usabilidade que têm sido usados para avaliar a usabilidade de aplicações Web e sua relação com o processo de desenvolvimento. Neste capítulo, é apresentada uma extensão do mapeamento apresentado em [12], reduzindo o seu escopo do ponto de vista da seguinte questão de pesquisa:

“Quais métodos de inspeção de usabilidade têm sido empregados por pesquisadores para avaliar artefatos Web e como estes métodos têm sido aplicados?”

Nesta revisão sistemática da literatura foram selecionados artigos sobre novos métodos de Inspeção de Usabilidade. Além disso, foram criadas sub-questões de pesquisa para analisar o atual estado dos Métodos de Inspeção de Usabilidade para aplicações *Web*. A análise dos resultados apresenta informações relevantes sobre os novos *UIMs* que têm surgido, em respeito a: (a) em que nível está sendo automatizado o processo de inspeção; (b) qual é o artefato *Web* mais comumente avaliado; e (c) quais métodos e/ou tecnologias são usados pelos *UIMs* como base teórica para executar o processo de inspeção de usabilidade.

Os resultados apresentados neste capítulo foram publicados no artigo: *Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension* [29], publicado no *Experimental Software Engineering Latin American Workshop (ESELAW) 2012*. Este artigo também foi selecionado para extensão no *CLEI Electronic Journal*.

O restante deste capítulo é apresentado a seguir: a Seção 2.2 introduz os conceitos relacionados a inspeções de usabilidade, os principais métodos de inspeção utilizados e apresenta a motivação para a execução da extensão do mapeamento sistemático. A Seção 2.3, por sua vez, apresenta as etapas para a execução da extensão do mapeamento sistemático em [12] sobre métodos de avaliação de usabilidade. A Seção 2.4 apresenta os resultados obtidos na extensão do mapeamento que respondem as questões de pesquisa propostas. Na Seção 2.5 são descritos resumidamente cada um dos *UIMs* de aplicações *Web* identificados, enquanto a Seção 2.6 discute as descobertas em relação aos *UIMs* e as suas implicações para a área acadêmica e a indústria. Finalmente a Seção 2.7 apresenta as considerações finais desta revisão da literatura.

2.2 Inspeções de Usabilidade

2.2.1 Avaliação de Usabilidade

Métodos de Avaliação de Usabilidade (*Usability Evaluation Methods - UEMs*) são procedimentos compostos por um conjunto bem definido de atividades que são usadas para avaliar a usabilidade do sistema [15]. Existem duas categorias de *UEMs*: teste com usuários e inspeções. Um teste com usuários é uma avaliação centrada no usuário, na qual métodos experimentais, observacionais e técnicas baseadas em perguntas podem ser usados para avaliar a usabilidade. O teste com usuários captura dados de uso de usuários reais enquanto estes fazem uso do produto (ou um protótipo) para completar

um conjunto definido de atividades. A análise dos resultados oferece informações que podem ser usadas para detectar problemas de usabilidade para melhorar o modelo de interação do sistema. Inspeções, por outro lado, são avaliações em que inspetores verificam a conformidade dos artefatos *Web* com aspectos de usabilidade. Os inspetores baseiam sua avaliação em *guidelines* que verificam o nível de concordância do sistema com os atributos de usabilidade. Com base na sua avaliação, os inspetores podem prever se ocorrerá ou não um problema de usabilidade.

Ao executar uma avaliação de usabilidade, as duas abordagens possuem vantagens e desvantagens. O teste com usuários é mais eficaz na identificação de problemas de usabilidade que afetam usuários finais [32]. No entanto, segundo Matera *et al.* [21], ele possui alto custo, pois requer muitos recursos para cobrir os diferentes perfis de usuário. Além disso, como requer toda ou pelo menos parte da implementação do sistema, é usado normalmente nas últimas etapas do processo de desenvolvimento. Métodos de inspeção de usabilidade, por outro lado, podem ser usados nas primeiras etapas do processo de desenvolvimento. Adicionalmente, esses métodos requerem menos recursos e conseqüentemente, podem diminuir os custos de encontrar problemas de usabilidade [32].

2.2.2 Métodos de Inspeção de Usabilidade

Segundo Matera *et al.* [21], os métodos mais representativos de inspeção de usabilidade são: a Avaliação Heurística de Nielsen [26] e o *Cognitive Walkthrough* [28]. Estes métodos são apresentados a seguir, assim como outros relacionados com o trabalho proposto.

Avaliação Heurística

A avaliação heurística é um método de avaliação de usabilidade proposto por Nielsen [26] baseado em diretrizes. O método auxilia o inspetor na avaliação de usabilidade visando diminuir os custos na avaliação de aplicações durante o processo de desenvolvimento. O processo de avaliação consiste de um conjunto de avaliadores que examinam a interface e categorizam defeitos de usabilidade baseado em uma lista de heurísticas. Estas heurísticas são um conjunto de regras ou diretrizes que visam descrever propriedades comuns de interfaces usáveis. A Tabela 2-1 apresenta as

heurísticas propostas por Nielsen [26] e os aspectos que um inspetor deve identificar ao avaliar a interface.

Tabela 2-1 Lista das heurísticas propostas por Nielsen [26]

Heurística	Descrição
1. Visibilidade do status do sistema	O sistema precisa manter os usuários informados sobre o que está acontecendo, fornecendo um feedback adequado dentro de um tempo razoável.
2. Compatibilidade do sistema com o mundo real	O sistema precisa falar a linguagem do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. Seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica.
3. Controle do usuário e liberdade	Os usuários frequentemente escolhem por engano funções do sistema e precisam ter claras saídas de emergência para sair do estado indesejado sem ter que percorrer um extenso diálogo. O sistema deve, portanto, prover funções “undo” e “redo”.
4. Consistência e padrões	Os usuários não precisam adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções de plataforma computacional.
5. Prevenção de erros	É melhor que o sistema possua um design cuidadoso o qual previna o erro antes dele acontecer.
6. Reconhecimento ao invés de relembração	O sistema deve tornar objetos, ações e opções visíveis. O usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.
7. Flexibilidade e eficiência de uso	Os usuários novatos se tornam peritos com o uso. O sistema deve prover aceleradores de forma a aumentar a velocidade da interação. O sistema deve permitir aos usuários experientes "cortar caminho" em ações frequentes.
8. Estética e design minimalista	Os diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.
9. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros	As mensagens de erro devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e construtivamente sugerindo uma solução.
10. Ajuda e documentação	É necessário prover ajuda e documentação embora seja melhor um sistema que possa ser usado sem documentação. Essas informações devem ser fáceis de

Heurística	Descrição
	encontrar, focalizadas na tarefa do usuário e não muito extensas.

Nielsen [26] recomenda que a avaliação de usabilidade seja efetuada por três a cinco avaliadores, devido ao fato que diferentes pessoas encontram diferentes problemas, o que melhora significativamente os resultados da avaliação. Durante o processo de inspeção cada avaliador percorre inicialmente a interface inspecionando os diferentes componentes do diálogo em busca de problemas. Se um problema for detectado, este será relatado em uma lista onde será associado com as heurísticas de usabilidade que foram violadas. Posteriormente, os inspetores podem classificar os problemas em: (a) não defeito, (b) cosmético, (c) leve, (d) grave, e (e) catastrófico.

Walkthrough Cognitivo

O *Walkthrough Cognitivo* (*Cognitive Walkthrough - CW*), também chamado de Percurso Cognitivo, é um método proposto por Polson *et al.* [28] que se baseia na técnica *walkthrough* de código. Segundo Matera *et al.* [21], o *Walkthrough Cognitivo* simula o processo de solução de problemas de um usuário; isto é, o que o usuário fará em determinadas circunstâncias de uso e porquê. Durante o processo de inspeção, os inspetores executam uma tarefa para verificar possíveis problemas de usabilidade. A avaliação é feita passo a passo, enquanto o *CW* guia os avaliadores na análise das ações que os usuários irão realizar para atingir um objetivo em uma determinada tarefa (ou cenário de uso). O *CW* identifica o relacionamento entre metas do usuário, ações e estados visíveis da interface da aplicação. O *Walkthrough Cognitivo* foca na compreensão e no aprendizado de uma aplicação por um usuário novato.

Segundo Polson *et al.* (1992), os itens necessários para realizar uma inspeção usando o *Walkthrough Cognitivo* são:

1. Uma especificação ou protótipo do sistema (bem detalhados);
2. Uma descrição da tarefa que o usuário deve executar no sistema, devendo esta ser uma tarefa representativa, que a maioria dos usuários irá fazer;
3. Uma lista escrita completa das ações necessárias para completar a tarefa com o sistema proposto;
4. Uma indicação de quem serão os usuários e que tipo de experiência e conhecimento os avaliadores podem assumir sobre eles.

Para executar uma avaliação, Polson *et al.* [28] sugerem que os inspetores façam as seguintes perguntas:

1. A ação correta é suficientemente evidente para o usuário?
2. O usuário será capaz de ligar a descrição da ação com o que está tentando fazer?
3. O usuário interpretará a resposta do sistema com a ação correta?

Durante o processo de inspeção os inspetores documentam as respostas obtidas para cada uma das perguntas para cada ação da lista apresentada. O processo de identificação de defeitos consiste em registrar as respostas negativas para qualquer uma das perguntas. Posteriormente, é possível indicar a severidade do problema encontrado.

2.2.3 Motivação para uma revisão sobre *UIMs*

Existem vários estudos sobre Métodos de Avaliação de Usabilidade que têm sido publicados [12]. Contudo, são poucos os que tratam do estado da arte dos Métodos de Avaliação de Usabilidade. Insfran e Fernandez [15] investigaram quais métodos têm sido empregados para avaliar a usabilidade de aplicações *Web* por meio da execução de um mapeamento sistemático. Mais tarde, Fernandez *et al.* [12] publicaram uma versão estendida do mapeamento sistemático adicionando novos estudos e uma análise mais aprofundada. O objetivo de ambas as publicações era condensar a informação sobre o uso dos novos *UEMs* e assim servir de base para empresas e pesquisadores que trabalham na área.

Com respeito a Métodos de Inspeção de Usabilidade, que são um subconjunto dos Métodos de Avaliação de Usabilidade, existem muitos trabalhos publicados que tratam de novos métodos. No entanto, não foram encontrados estudos sobre a caracterização desses novos métodos de inspeção de usabilidade. Como consequência, foi feita uma extensão do estudo realizado por Fernandez *et al.* [12], selecionando os artigos descrevendo novos *UIMs* e analisando-os rigorosamente. A análise inclui novos critérios de extração de dados, o que permite oferecer informação útil referente aos novos *UIMS* e a sua aplicação.

2.3 Procedimento de Execução da Extensão do Mapeamento Sistemático

A extensão do mapeamento sistemático foi baseada no mapeamento executado por Fernandez *et al.* [12], cujo trabalho responde a seguinte questão de pesquisa:

“Quais métodos de avaliação de usabilidade têm sido empregados por pesquisadores para avaliar a usabilidade de artefatos Web, e como estes métodos têm sido empregados?”

Nesse estudo, é analisado e resumido o conhecimento respeito a métodos de avaliação de usabilidade de aplicações Web. Contudo, a análise dos métodos de inspeção de usabilidade de aplicações Web é muito genérica e, como consequência, é preciso de uma descrição mais detalhada para entender como os UIMs como são empregados.

A Figura 2-1 apresenta as quatro etapas para a execução da extensão do mapeamento sistemático: seleção de estudos primários, estratégia de extração de dados, execução e documentação. Nesta seção são discutidas as etapas prévias à documentação dos resultados.

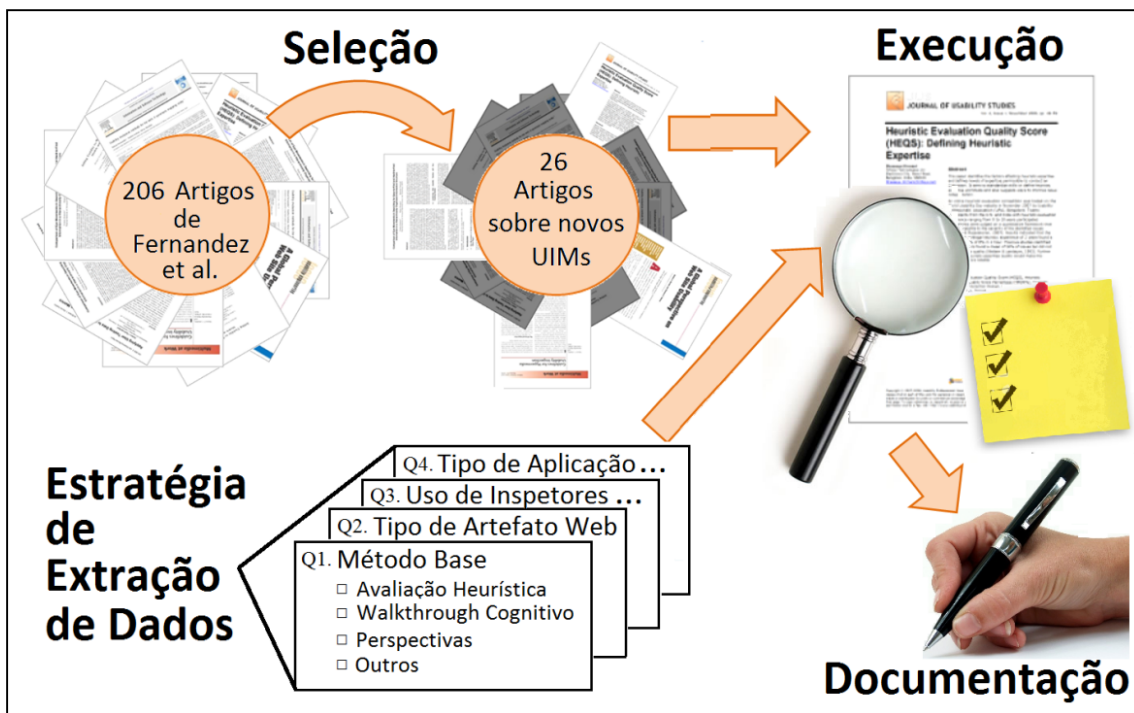


Figura 2-1 Etapas do processo de execução da extensão do mapeamento sistemático com base no mapeamento em [12]. Figura adaptada de [29].

2.3.1 Seleção de Estudos Primários

Em Fernandez *et al.* [12] foram considerados um total de 206 artigos que discutem a utilização de UEMs. Para responder à sua questão de pesquisa, os autores do mapeamento sistemático criaram sete sub-questões de pesquisa, cada uma pareada com

um conjunto possível de respostas. A Tabela 2-2 mostra as sub-questões de pesquisa usadas para dividir os estudos encontrados em categorias com a finalidade de classificá-los.

O objetivo com essa extensão do mapeamento sistemático é analisar o uso atual dos novos *UIMs* no domínio *Web* ao responder a seguinte questão de pesquisa:

“Quais novos métodos de inspeção de usabilidade têm sido empregados por pesquisadores para avaliar artefatos Web, e como estes têm sido empregados?”

Uma análise dos termos em negrito da Tabela 2-2 permite observar, que os artigos com respostas: “novo” e “inspeção”, para as perguntas Q1 e Q3 respectivamente estão relacionados com a questão de pesquisa da extensão do mapeamento sistemático. Conseqüentemente, de um total de 206 artigos, citados em Fernandez *et al.* [12] foram selecionados aqueles descrevendo a aplicação de novos métodos de inspeção.

Tabela 2-2 Sub-questões de pesquisa do mapeamento sistemático de Fernandez *et al.* [12] e suas possíveis respostas.

Sub-questões de Pesquisa	Possíveis Respostas	
Q1. Origem do UEM empregado	a) Novo	b) Existente
Q2. Definição de Usabilidade do UEM empregado	a) Standard	b) Ad-Hoc
Q3. Tipos de UEMs empregados	a) Teste com Usuários c) Simulação e) Questionário	b) Inspeção d) Modelagem Analítica
Q.4 Validação Experimental do UEM	a) Questionário c) Experimento	b) Estudo de Caso d) Não
Q5. Tipo de Avaliação Executada pelo UEM	a) Manual	b) Automatizado
Q6. Fases nos que o UEM pode ser aplicado.	a) Requisitos c) Implementação	b) Design
Q7. Feedback fornecido pelo UEM	a) Sim	b) Não

Devido à abrangência da questão de pesquisa para conseguir descrever como os *UIMs* estão sendo aplicados, também foi dividida em sub-questões para ser respondida com mais detalhe. A Tabela 2-3 mostra estas sub-questões de pesquisa e sua motivação.

Do conjunto inicial de artigos obtidos de Fernandez *et al.* [12] foram selecionados somente aqueles descrevendo detalhadamente UIMs. Consequentemente, foram descartados os estudos que se encaixassem em qualquer um dos critérios de exclusão definidos a seguir:

- Artigos apresentado problemas de usabilidade e nenhuma metodologia para identificá-los.
- Artigos descrevendo apenas ideias de pesquisa.
- Artigos apresentado técnicas sem descrever o processo de execução das mesmas.

Tabela 2-3 Sub-questões de pesquisa para a execução da extensão do mapeamento sistemático e sua motivação.

Sub-questão de Pesquisa	Motivação
Q1. Método Base	Verificar se os Métodos de Inspeção de Usabilidade de Aplicações <i>Web</i> têm sido desenvolvidos considerando métodos reconhecidos ou se estão usando novos embasamentos teóricos.
Q2. Tipo de Artefato Avaliado	Encontrar qual é o artefato mais comumente avaliado por Inspeções de Usabilidade de Aplicações <i>Web</i> .
Q3. Tipo de Aplicação <i>Web</i> avaliada pelo Método de Inspeção de Usabilidade	Descobrir se os Métodos de Inspeção de Usabilidade de Aplicações <i>Web</i> estão sendo desenvolvidos para encontrar problemas genéricos de usabilidade ou problemas de um tipo específico de aplicação <i>Web</i> .
Q4. Uso de Inspetores no Processo de Inspeção	Averiguar se os Métodos de Inspeção de Usabilidade de aplicações <i>Web</i> estão sendo automatizados ao ponto de não precisar de inspetores para ser executados.

2.3.2 Estratégia de Extração de Dados

Com a finalidade de extrair dados para posterior análise, foram usadas as perguntas correspondentes a cada sub-questão de pesquisa definida. Esta estratégia garante que os estudos selecionados sejam classificados e categorizados de forma objetiva. As possíveis respostas para cada uma das sub-questões de pesquisa são explicadas a seguir:

Com respeito à questão Q1 (Método Teórico Base), um artigo pode ser classificado em uma das seguintes respostas: (a) Avaliação Heurística: se o método apresentado é um UIM baseado na Avaliação Heurística de Nielsen [26]; (b) Walkthrough Cognitivo: se o método apresentado é um *UIM* baseado no Walkthrough Cognitivo definido por Polson *et al.* [28]; (c) Baseado em Perspectivas: se o método apresentado é um UIM baseado em diferentes Perspectivas (como o método de Zhang *et*

al. [41] Perspective Based Usability Evaluation); e (d) Nova Base: se o método apresentado se baseia em uma nova abordagem.

Com respeito à questão Q2 (Tipo de Artefato Avaliado), um artigo pode ser classificado em uma das seguintes respostas: (a) HTML: se o método apresentado faz uso de análise de código para encontrar problemas de usabilidade; (b) Modelo: se o método apresentado faz uso de análise de modelos para encontrar problemas de usabilidade; e (c) Aplicação/Protótipo: se o método apresentado faz uso de análise de protótipos e/ou aplicações para encontrar problemas de usabilidade.

Com respeito à questão Q3 (Tipo de Aplicação *Web* avaliada pelo Método de Inspeção), um artigo pode ser classificado em uma das seguintes respostas: (a) Genérica: se o método apresentado pode ser aplicado a qualquer tipo de aplicação *Web*; e (b) Específico: se o método apresentado avalia tipos específicos de aplicações *Web*.

Com respeito à questão Q4 (Uso de inspetores no processo de Inspeção), um artigo pode ser classificado em uma das seguintes respostas: (a) Sim: se o método apresentado precisa da intervenção de inspetores para executar a inspeção; e (b) Não: se o método apresentado é totalmente automatizado e pode executar uma avaliação de usabilidade sem a intervenção de inspetores experientes.

2.3.3 Execução

Do conjunto inicial de 206 artigos de Fernandez *et al.* [12], foram selecionados 37 artigos usando as respostas às questões Q1 e Q3 disponíveis no mapeamento sistemático. No entanto, 5 artigos não estavam disponíveis para download e o conjunto inicial teve de ser reduzido para 32.

Após a leitura cuidadosa de cada estudo, foram descartados 6 artigos por se encaixarem em um dos critérios de exclusão definidos acima. Os 26 estudos selecionados para análise nesta extensão estão listados na Tabela 2-4. Maiores detalhes sobre o ano de publicação do artigo e a conferência onde foi publicado podem ser encontrados no Apêndice A. A estratégia de extração de dados definida na subseção anterior foi usada para identificar o estado atual dos *UIMs* de aplicações *Web*. Os resultados são discutidos na Seção 2.4.

Tabela 2-4 Código, título, autores e veículo de publicação dos artigos analisados na extensão do mapeamento sistemático.

Código	Título	Autores	Fonte
S01	<i>"Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology"</i>	Allen, M.; Currie, L.; Bakken, S.; Patel, V.; Cimino, J.	SD
S02	<i>"An HTML analyzer for the study of web usability"</i>	Rios, D.; Vazquez, I.; Rey, E.; Bonillo, V.; Del Rio, B.	IEEEEx
S03	<i>"A tool to support usability inspection"</i>	Ardito, C.; Lanzilotti, R.; Buono, P.; Piccinno, A.	ACM
S04	<i>"Adding Usability to Web Engineering Models and Tools"</i>	Atterer, R.; Schmidt, A.	ICWE
S05	<i>"Context-driven assessment of commercial Web sites"</i>	Basu, A.	IEEEEx
S06	<i>"Cognitive walkthrough for the web"</i>	Blackmon, M.; Polson, P.; Kitajima, M.; Lewis, C.	ACM
S07	<i>"Repairing usability problems identified by the cognitive walkthrough for the web"</i>	Blackmon, M.; Kitajima, M.; Polson, P.	ACM
S08	<i>"Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs"</i>	Blackmon, M.; Kitajima, M.; Polson, P.	ACM
S09	<i>"Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+"</i>	Bolchini, D.; Garzotto, F.	IWWUA
S10	<i>"Will World Wide Web user interfaces be usable?"</i>	Burton, C.; Johnston, L.	IEEEEx
S11	<i>"Applying user testing data to UEM performance metrics"</i>	Chattratchart, J.; Brodie, J.	ACM
S12	<i>"Web usability inspection technique based on design perspectives"</i>	Conte, T.; Massollar, J.; Mendes, E.; Travassos, G.	IEEEEx
S13	<i>"Guidelines for hypermedia usability inspection"</i>	Costabile, M.; Matera, M.	IEEEEx
S14	<i>"Recoverability Walkthrough: An Alternative to Evaluate Digital Inclusion Interfaces"</i>	Filgueiras, L.; Martins, S.; Tambascia, C.; Duarte, R.	IEEEEx
S15	<i>"Identifying Cultural Markers for Web Application Design Targeted to a Multi-Cultural Audience"</i>	Fraternali, P.; Tisi, M.	ICWE

Código	Título	Autores	Fonte
S16	<i>"Comparison of eye movements in searching for easy-to-find and hard-to-find information in a hierarchically organized information structure"</i>	Habuchi, Y.; Kitajima, M.; Takeuchi, H.	ACM
S17	<i>"Heuristic Evaluation Quality Score (HEQS): Defining Heuristic Expertise"</i>	Kirmani, S.	JUS
S18	<i>"Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems"</i>	Molina, F.; Toval, A.	SD
S19	<i>"Ontology driven definition of a usability model for second generation portals"</i>	Moraga, M.; Calero, C.; Piattini, M.	ICWE
S20	<i>"UWIS: An assessment methodology for usability of web-based information systems"</i>	Oztekin, A.; Nikov, A.; Zaim, S.	SD
S21	<i>"Automatic reconstruction of the underlying interaction design of web applications"</i>	Paganelli, L.; Paterno, F.	ACM
S22	<i>"Hypermedia, the Web and Usability issues"</i>	Paolini, P.	IEEEEx
S23	<i>"A comprehensive model for Web sites quality"</i>	Signore, O.	IEEEEx
S24	<i>"Web 2.0: extending the framework for heuristic evaluation"</i>	Thompson, A.J.; Kemp, E.A.	ACM
S25	<i>"Evaluating Web usability with MiLE+"</i>	Triacca, L.; Inversini, A.; Bolchini, D.	IEEEEx
S26	<i>"Automated Evaluation of Web Usability and Accessibility by Guideline Review"</i>	Vanderdonckt, J.; Beirekdar, A.; Noirhomme-Fraiture, M.	ICWE

2.4 Resultados

Os resultados da classificação de estudos primários e o seu resumo são mostrados na Tabela 2-5. Estes resultados finais foram calculados somando o número total de respostas para cada uma das sub-questões de pesquisa. Deve ser considerado que as perguntas Q1 e Q2 não são exclusivas, ou seja, um estudo pode ser classificado em uma ou mais respostas e, como consequência, o somatório das porcentagens pode ser maior que 100%.

Tabela 2-5 Resultados da classificação de estudos primários sobre UIMs para a Web.

		Artigo																								Artigos Totais	Porcentagem (%)		
		01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24			25	26
Q01	A	x									x	x	x					x			x							7	26.92
	B						x	x	x						x		x											5	19.23
	C									x			x												x		x	4	15.38
	D		x	x	x	x				x				x		x			x	x	x	x	X	x		x	x	15	57.69
Q02	A		x																								x	4	15.38
	B				x															x							x	4	15.38
	C	x		x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	X		x	x		20	76.92	
Q03	A		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	X	x		x	x	23	88.46	
	B	x				x																				x		3	11.54
Q04	A	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	X	x	x	x		24	92.31	
	B		x																								x	2	7.69

Categorização de Estudos Primários

Q1.- Método Teórico Base:

(a) Avaliação Heurística (b) Walkthrough Cognitivo (c) Baseado em Perspective (d) Outra Base

Q2.- Tipo de Artefato Avaliado:

(a) Análise de Código HTML (b) Análise de Modelo (c) Análise de Application/Protótipo

Q3.- Tipo de Application Web Avaliada pelo Método de Inspeção:

(a) Genérico (b) Específico

Q4.- Uso de Inspetores no Processo de Inspeção:

a) Sim (b) Não

Nas próximas subseções é apresentada a análise dos resultados para cada uma das sub-questões de pesquisa, descrevendo como os métodos analisados têm sido utilizados. Cada artigo da Tabela 2-4 será referenciado pelo seu código.

2.4.1 Método Teórico Base

Os resultados para a sub-questão de pesquisa Q1 (Método Teórico Base) revelaram que, com respeito a métodos de inspeção comumente usados, ao redor de 27% dos artigos avaliados se basearam na avaliação heurística de Nielsen [26], ao redor de 19% se baseou no Walkthrough Cognitivo de Polson *et al.* [28], e ao redor de 15% se basearam em perspectivas.

Os estudos baseados na Avaliação Heurística de Nielsen [26], focaram principalmente em melhorar a descrição ou a autoexplicação de como ou em que situações cada heurística pode ser aplicada. Os artigos S11 e S24 oferecem uma série de exemplos que auxiliam na identificação de problemas de usabilidade ao usar heurísticas.

Os artigos S12 e S20 aperfeiçoaram a lista original de Heurísticas de Nielsen ao combiná-las com outros métodos.

Com respeito ao uso do Walkthrough Cognitivo, dois novos métodos de inspeção foram desenvolvidos: o *Cognitive Walkthrough for the Web* de Blackmon *et al.* descrito nos artigos S06, S07 e S08; e o *Recoverability Walkthrough* descrito no artigo S14. Adicionalmente, o artigo S16 valida como o método *Eye Tracking* e o Walkthrough Cognitivo podem ser combinados em um estudo executado por Hubuchi *et al.*

Os estudos descritos em S09, S12, S23 e S25 fazem uso de perspectivas para apoiar inspetores na identificação de problemas de usabilidade de diferentes aspectos de usabilidade durante o processo de inspeção. Esta base teórica foi utilizada em conjunto com outras bases para melhorar o desempenho das técnicas propostas.

O 58% de artigos restantes reportaram técnicas que estão sendo propostas com foco em aplicações *Web*. A maioria dos estudos descreve técnicas baseadas em heurísticas sugeridas especificamente para o domínio *Web*. Outra abordagem é a transformação de princípios de usabilidade encontrados em guias como a ISO 9241-11 [16].

Os resultados podem indicar que novas abordagens estão sendo sugeridas. No entanto, muitos dos *UIMs* propostos ainda fazem uso de técnicas usadas para avaliar qualquer tipo de aplicação, e as estão adaptando para o domínio *Web*. De fato, alguns dos artigos selecionados (S12, S20 e S23), que fazem uso de métodos adaptados, combinam uma ou duas abordagens para melhorar o desempenho do inspetor durante o processo de avaliação.

2.4.2 Tipo de Artefato Avaliado

Os resultados da sub-questão de pesquisa Q2 (Tipo de Artefato Avaliado) revelaram que a maioria dos *UIMs* analisa Aplicações e ou Protótipos. Ao redor de 77% dos artigos avaliados descreveram *UIMs* que fazem uso de protótipos e/ou sistemas funcionais ou ainda protótipos em papel. Os inspetores podem executar uma avaliação ao analisar a interação oferecida pelo protótipo ou produto enquanto executam tarefas no mesmo. Estes resultados podem indicar que, a não ser que o protótipo seja feito em papel, a maioria das técnicas de inspeção de usabilidade estão sendo aplicadas nas últimas etapas do processo de desenvolvimento onde pelo menos parte do sistema foi desenvolvida.

O restante dos artigos está dividido em análise de código HTML (15%) e análise de modelos (15%). Os artigos S02, S21, S23 e S26 descrevem técnicas automatizadas onde o código HTML pode ser analisado para identificar problemas de usabilidade com base em guias e padrões de usabilidade. Os artigos selecionados mostram que a análise de código HTML é usada, na maioria dos casos, nas últimas etapas do processo de desenvolvimento, quando o código real do sistema está disponível para inspeção. A automatização da análise de código HTML pode ajudar a reduzir os custos deste tipo de inspeção. No entanto, isso não implica que a correção dos problemas de usabilidade terá um custo baixo, pois, eles serão encontrados nas últimas etapas do processo de desenvolvimento. Os estudos S04, S18, S21 e S23 mostram *UIMs* que fazem uso de modelos para identificar problemas. Modelos ajudam na representação da interação humano computador enquanto o inspetor verifica se o modelo está de acordo com as regras de interação do domínio *Web*. Este tipo de inspeções pode ajudar a encontrar problemas de usabilidade nas primeiras etapas do processo de desenvolvimento, podendo reduzir assim o custo de corrigi-los. Contudo, segundo os estudos selecionados, este tipo de *UIM* ainda precisa de inspetores experientes para analisar e avaliar artefatos *Web*.

2.4.3 Tipo de Aplicação Web Avaliada pelo método de Inspeção

Os resultados para a sub-questão de pesquisa Q3 (Tipo de Aplicação *Web* avaliada pelo processo de inspeção) revelou que ao redor de 88% dos estudos selecionados podem ser aplicados a qualquer tipo de aplicação *Web*. Por exemplo, os artigos S06, S12 e S14 descrevem *UIMs* que adaptam métodos genéricos de inspeção de usabilidade ao domínio *Web*. Os estudos S18, S19 e S25 são exemplos de técnicas que criaram suas próprias características de usabilidade, mas ainda continuaram flexíveis para avaliar qualquer tipo de aplicação *Web*.

O restante dos artigos (S01, S05 e S24), cerca de 12% dos estudos selecionados, propôs métodos de inspeção para identificar problemas de usabilidade em tipos específicos de aplicações *Web*. Allen *et al.* (S01) descreveu uma técnica baseada em protótipos em papel para avaliar a usabilidade de aplicações *Web* médicas. Basu (S05) propôs um novo framework para avaliar aplicações *Web* de e-commerce. Esta abordagem focava no contexto de uso da aplicação para encontrar problemas de usabilidade. No artigo S24, Thompson e Kemp avaliaram a usabilidade de aplicações *Web* desenvolvidas no contexto da *Web 2.0*.

2.4.4 Uso de Inspectores no processo de inspeção

Os resultados para a sub-questão de pesquisa Q4 (Uso de Inspectores no Processo de Inspeção) revelaram que ao redor de 92% dos estudos selecionados precisam da intervenção de um inspetor para executar uma inspeção de usabilidade. O restante dos artigos (8%) não fez uso de inspetores. Uma análise demonstra que existe uma relação entre o artefato avaliado pelo *UIM* e o grau de automatização dos inspetores. Foi possível identificar que inspeções que faziam uso de análise de código HTML estavam sendo totalmente automatizadas, enquanto outros métodos ainda faziam uso de inspetores. Os *UIMs* propostos em S21 e S23 avaliavam outros artefatos além de analisar o código HTML, conseqüentemente, faziam uso de inspetores e não foram totalmente automatizados.

2.5 Resumo dos Métodos de Inspeção de Usabilidade de Aplicações Web

Após a leitura e análise de cada artigo, foi identificado um total de 21 novas técnicas de inspeção de usabilidade de aplicações *Web*. É importante ressaltar que apesar de haver um total de 26 artigos selecionados, 5 deles falavam de técnicas já citadas, seja evoluindo-as ou realizando estudos experimentais para validá-las. Nesta seção, será apresentada brevemente, cada uma das técnicas identificadas. Estas técnicas foram ordenadas segundo a data de publicação dos artigos.

Burton e Jhonston adaptaram a Avaliação Heurística [26], no artigo S10, para propor questões e usá-las para encontrar problemas de usabilidade em aplicações *Web*. Os autores também executaram um estudo experimental onde as questões foram usadas para realizar uma inspeção. Os resultados deste experimento foram armazenados em um repositório para permitir seu uso posterior.

No artigo S13, Matera e Costabile propuseram o *Abstract Task (AT) Based Inspection*, uma técnica que guia os inspetores com base no Modelo de *Design* de Hipermedia [14] que sugere objetos de foco em uma avaliação *Web*. A *AT* sugere o uso de tarefas abstratas compostas por: título, código de classificação, intenção, descrição, output e exemplo. Durante uma avaliação, um inspetor realiza as tarefas relacionadas com a aplicação avaliada e anota suas observações.

O artigo S06 de Blackmon descreve o *Walkthrough* Cognitivo para a *Web* (*Cognitive Walkthrough for the Web - CWW*). Este método foi descrito também no S07 e S08 e é uma evolução do *Walkthrough* Cognitivo. O *CWW* divide a interação em duas

fases: atenção e seleção da ação. A avaliação segue o mesmo processo que a do *Walkthrough Cognitivo*, porém utiliza duas perguntas diferentes: Q2a) *O usuário será capaz de conectar a sub-região da página com o seu objetivo, usando as informações disponibilizadas e o seu entendimento do layout?* e Q2b) *O usuário será capaz de relacionar o objetivo com o widget correto, usando as informações da página?* Caso seja identificado um problema, os links confusos, competitivos e/ou problemáticos serão renomeados.

Em S21, Paganelly e Paterno descrevem o *Automatic Reconstruction of the Underlying Interaction Design of Web Applications*, a produção automática do modelo de tarefas com base em código HTML. A técnica avalia o código em busca de elementos de interação (links, botões, entre outros) e grupos de elementos (campos, formas, etc) e o transforma em um modelo de tarefas. Este modelo é usado por outro sistema que gera um relatório sobre as diferenças entre o layout e a forma em que os usuários realmente usam o sistema. Este relatório é usado por especialistas de usabilidade para tomar decisões sobre mudanças no design da aplicação.

Basu propôs um framework para medir a usabilidade da *Web* no artigo S05. Este *UIM* foi desenvolvido para aplicações e-commerce. Segundo Basu (S05), aplicações e-commerce possuem três dimensões que podem ser usadas para avaliar a usabilidade: (a) contexto do e-business, (b) processos online apoiados, e (c) fatores de definição de usabilidade. O método de inspeção ajuda a identificar os objetivos dos processos da aplicação *Web*. Cada fator de usabilidade deve ser medido de acordo com os processos da aplicação. Finalmente, os problemas de usabilidade são identificados com base nas características que a aplicação *Web* possui para apoiar esses processos.

Em S11, Chattratchart e Brodie propuseram a *Avaliação Heurística Plus*. Este método instancia a *Avaliação Heurística*. Os autores executaram uma inspeção de uma aplicação *Web* de centros comerciais. Para isso, mediram três aspectos específicos: (a) correção, que é a taxa entre o número de defeitos reais encontrados, dividido pelo total de defeitos reais; (b) validade, que é a taxa de defeitos reais encontrados, dividido pelo total de defeitos encontrados; e (c) eficácia, que é o produto da correção com a validade.

Já Vanderdonckt *et al.* (S26), propuseram uma avaliação automática de usabilidade e acessibilidade na *Web*. Os autores descreveram uma ferramenta que analisa código HTML de forma automática em busca de problemas de usabilidade. A ferramenta deve ser usada em conjunto com outras técnicas para verificar a satisfação do usuário, consistência e organização da informação.

O artigo S04 trata sobre um validador de usabilidade baseado em modelos. Neste artigo, os autores, Atterer e Schmidt, propuseram incluir atributos de usabilidade nos modelos *Web* para que pudessem ser avaliados por ferramentas de validação. Para isso, os autores sugerem a inclusão dos atributos: tempo, propósito e público alvo. Estes atributos são medidos e um conjunto de *guidelines* é verificado em busca de problemas de usabilidade.

No artigo S23, Signore propõe um modelo compreensivo para a qualidade de *Sites Web*. Este *UIM* é um método híbrido que faz uso de código HTML e análise de modelos. O modelo usa cinco perspectivas para avaliar a usabilidade: corretude, apresentação, conteúdo, navegação e interação. É importante ressaltar que a perspectiva de corretude está diretamente relacionada com a qualidade do código, enquanto as outras perspectivas estão relacionadas com a opinião dos usuários. A perspectiva de corretude é verificada avaliando o código HTML, enquanto as outras perspectivas são avaliadas com base nas lições aprendidas das inspeções realizadas.

Os estudos apresentados em S09 e S25 apresentam o método *Milano Lugano Evaluation Method (MiLE+)*. Bolchini e Garzotto propuseram esse método para avaliar a usabilidade dos sistemas *Web* a partir de duas perspectivas: perspectiva técnica e perspectiva de experiência do usuário. Para avaliar a perspectiva técnica, os inspetores devem usar um total de 81 heurísticas divididas em quatro aspectos: navegação, conteúdo, tecnologia, e design de interface. Com relação à perspectiva de experiência do usuário, os inspetores devem usar um total de 20 indicadores divididos em três aspectos: experiência em conteúdo, experiência em navegação e experiência em fluxo operacional. Ao utilizar o *MiLE+*, os inspetores criam cenários e avaliam os aspectos dependentes da aplicação com os indicadores, e os aspectos independentes com as heurísticas.

O *Paper-Based Heuristic Evaluation* é um *UIM* especificamente criado para a avaliação de aplicações *Web* médicas. Em S01, Allen *et al.* sugeriram a utilização de cinco heurísticas: consistência, correspondência, minimalismo, memória e linguagem. Estas heurísticas são o resultado da combinação das heurísticas de Nielsen [26] e as regras de ouro de Shneiderman [34]. O processo de inspeção deste método é similar ao processo de inspeção da avaliação heurística, no entanto, são produzidos *mockups* ou *print screens* das telas para realizar a avaliação.

Moraga *et al.* (S19) propuseram a Ontologia para a Medição de *Software*. Esta ontologia se baseia na recopilação de termos usados relacionados com usabilidade. Em

S19, os autores afirmam que existem dois tipos de aplicações *Web*: (a) a primeira geração com páginas *Web* estáticas com estrutura e funcionalidades fixas; e (b) a segunda geração com aplicações compostas por *portlets*. *Portlets* são mini aplicações com funcionalidades limitadas e que juntas podem chegar a compor um sistema maior. A avaliação de usabilidade consiste em escolher quais *portlets* devem compor a aplicação com base no seu nível de usabilidade. Para medir a usabilidade dos *portlets*, os autores propuseram quatro indicadores: entendimento, capacidade de aprender, customização e observância.

No artigo S15, Fraternaly e Tisi projetaram uma metodologia para identificar marcadores culturais de aplicações *Web*. Este UIM tem por objetivo apoiar a identificação de fatores que podem ter um efeito nos usuários finais com base no seu background cultural. Para fazer isso, os autores descreveram como executar uma série de estudos para classificar as aplicações *Web* e identificar os marcadores culturais um impacto perceptível sobre a usabilidade. Os autores descobriram que a organização, cores e padrões de navegação das aplicações *Web* devem refletir a cultura do usuário, ou como consequência, a aplicação não será usável. A inspeção consiste na identificação e validação dos marcadores para usar as informações e sugerir melhorias no desenvolvimento das próximas aplicações.

Habuchi *et al.* (S16) sugeriram a combinação do *CWW* (S06) com o *Eye Tracking* [7]. O *Eye Tracking* é o processo de medição do ponto de visão ou o movimento do olho em relação à cabeça. Esta medição é feita para obter dados relacionados com o efeito produzido ao modificar a apresentação visual da informação. No método de Habuchi *et al.*, sugere-se utilizar o *CWW* para identificar e avaliar os links confusos ou fracos. Após esta etapa *Eye Tracking* serve para verificar o tempo perdido ao encontrar a melhor solução em vez de entendê-la.

No artigo S17, Kirmani propôs a Pontuação de Qualidade da Avaliação Heurística. Este método é baseado na Avaliação Heurística [26] e a evolui em termos de categorização dos defeitos. O artigo trata da grande dificuldade dos inspetores em classificar de forma correta os defeitos encontrados. Apesar de conseguirem encontrar defeitos, é difícil classificá-los em catastróficos, graves, leves ou cosméticos. Com a finalidade de solucionar esse problema, o autor sugere a utilização de duas perspectivas: usuário e ambiente.

Alonso-Rios *et al.* (S02) propuseram um analisador de código HTML. Este analisador é baseado nos problemas relatados por Nielsen [26]. Segundo os autores,

apesar de ser um analisador de HTML, o método inclui aspectos de usabilidade não considerados por outros analisadores. Por exemplo, o analisador verifica se os links estão corretamente destacados para garantir que os usuários os vejam. Além disso, a proposta conta com apoio ferramental que aponta os problemas de usabilidade e faz sugestões de correção.

A técnica *Web Design Perspectives* - *WDP* foi descrita no artigo S12. Esta técnica foi proposta por Conte *et al.* e combina duas abordagens: o uso de perspectivas e heurísticas. A técnica *WDP* foi baseada na Avaliação Heurística e agrupa suas heurísticas com base em três principais perspectivas: navegação, conceituação e apresentação. O processo de inspeção de usabilidade consiste na seleção de tarefas representativas da aplicação *Web* a ser avaliada. Após esta etapa, os inspetores executam as tarefas usando as heurísticas de cada perspectiva para encontrar problemas de usabilidade. Posteriormente uma equipe combinará todas as listas para gerar uma lista contendo todos os problemas relatados, decidirá quais desses problemas são efetivamente problemas de usabilidade.

No artigo S14, Filgueiras *et al.* propuseram o *Recoverability Walkthrough* (*RW*). Como o *CWW* (S06), o *RW* também se baseia no Walkthrough Cognitivo [28]. Este método adiciona mais duas perguntas ao conjunto de perguntas proposto no em [28]: (a) *Se o usuário não fizer as ações corretas, será possível para ele identificar que está indo no caminho errado?* e (b) *Se o usuário perceber que está indo no caminho errado, será possível para ele retornar ao estado antes do erro?* Antes do processo de inspeção, os usuários finais deverão ser modelados e os inspetores deverão garantir que a interação seja a melhor possível para assim melhorar a usabilidade do sistema. Neste método, os inspetores não fazem sugestões de melhoria, ao contrário são gravados e os designers tomam decisões com esta informação.

O artigo S18 de Molina e Troval descreve um método para integrar requisitos de usabilidade aos requisitos da aplicação *Web*. O UIM é chamado de *Model Driven Engineering of Web Information Systems* e se baseia no processo “*weaving*” que permite a combinação do modelo de navegação e requisitos de usabilidade com o modelo de requisitos. Este método permite a verificação de quatro aspectos de usabilidade: definição de níveis importantes, distância entre nós, restrições de conectividade e restrições de navegação. Quando um novo meta-método é obtido através do processo “*weaving*”, os inspetores usam um total de 50 métricas para avaliar se o modelo atinge os requisitos de usabilidade.

Finalmente, Oztekin *et al.* (S20) propõe o método *UWIS* (*Usability for Web Information Systems – Usabilidade para Sistemas de Informação Web*). Neste *UIM*, a usabilidade pode ser medida considerando três dimensões: eficiência, eficácia e satisfação. No entanto, para atingir melhores resultados, estas dimensões devem ser usadas em conjunto com um *checklist* de qualidade. O método *UWIS* sugere que é possível identificar o relacionamento entre a qualidade e os fatores de usabilidade. Portanto, utiliza as respostas a 27 questões sugeridas no método, para realizar uma análise estatística. Esta análise permite identificar que problemas possuem o maior e menor efeito negativo sobre a aplicação *Web* avaliada. Para realizar uma inspeção, o inspetor deve responder às perguntas e usar o Modelo de Equação Estrutural [17] para identificar os relacionamentos entre qualidade e fatores de usabilidade.

Na seguinte seção, serão relacionadas as descobertas da extensão do mapeamento sistemático com as suas respectivas implicações para a academia e a indústria.

2.6 Discussão dos Resultados da Extensão do Mapeamento Sistemático

2.6.1 Principais Descobertas

O objetivo desta revisão da literatura foi avaliar e examinar o atual estado dos métodos de inspeção de usabilidade que têm surgido para avaliar artefatos *Web*. As principais descobertas deste estudo secundário são as seguintes:

1. Métodos de Inspeção de Usabilidade de aplicações *Web* estão surgindo a partir de: (a) a evolução de métodos genéricos de inspeção de usabilidade ao adaptá-los para o domínio *Web*; e (b) criando novos critérios específicos para a *Web*. No entanto, como esses métodos fazem uso de dimensões de usabilidade diferentes dependendo da base teórica utilizada, nenhum deles pode ser usado para avaliar todos os tipos de aplicações *Web*. Por conseguinte, uma combinação dos métodos pode ajudar a aprimorar os resultados da avaliação.
2. Existe um grande número de *UIMs* para aplicações *Web* genéricas comparado com o número de *UIMs* para aplicações específicas. *UIMs* genéricos focam em encontrar problemas de usabilidade que se aplicam a qualquer aplicação *Web*, mas a maioria deles não oferece *feedback* sobre como tratar os problemas de usabilidade detectados. Por outro lado, *UIMs* que avaliam aplicações *Web* específicas oferecem informações aos inspetores sobre aquele tipo de aplicação.

Estes dados permitem à indústria e aos pesquisadores focar em aspectos de usabilidade que afetam um determinado tipo de aplicação, mas que nem sempre podem ser aplicados a todos os tipos de aplicações *Web*.

3. A automatização do processo de inspeção de técnicas envolvendo julgamento e interação humana ainda não está sendo feita. Como consequência, técnicas que fazem uso de análise de modelos e protótipos não estão sendo automatizadas, porém, estas técnicas estão sendo apoiadas por ferramentas que ajudam a diminuir o esforço durante a inspeção. Por outro lado, UIMs que fazem uso de análise de código HTML para encontrar inconsistências de padrões e cores já estão sendo totalmente automatizados. Mesmo assim, a avaliação realizada por este tipo de UIMs não cobre todos os aspectos de usabilidade que uma técnica fazendo uso de inspetores chegaria a cobrir.

2.6.2 Implicações para a Academia e a Indústria

Os resultados dessa revisão sistemática têm implicações para ambas a indústria e a academia. Nesta subseção serão discutidas sugestões para: (a) pesquisadores planejando estudos ou sugestões de técnicas de inspeção de usabilidade; e (b) empresas de desenvolvimento de software que trabalham com aplicações *Web* e estão interessadas em integrar UIMs no seu processo de desenvolvimento de forma efetiva.

Para os pesquisadores, uma análise mais detalhada dos UIMs tem permitido identificar uma inconsistência no uso de nomes, categorias e inclusão de aspectos de usabilidade. Aspectos de usabilidade são dimensões usadas por inspetores para identificar problemas de usabilidade. Durante a revisão foi descoberto que a maioria dos UIMs foca em diferentes aspectos que nem sempre estão presentes em todas as técnicas de inspeção. As novas propostas de UIMs devem considerar todos os aspectos de usabilidade definidos a fim de incluí-los, e não duplicá-los ou confundi-los. Esta estratégia pode contribuir na criação de UIMs que permitam realizar avaliações mais consistentes e completas.

Os resultados demonstram uma relação entre o tipo de artefato avaliado e o grau de automatização das UIMs que os avaliam (ver Tabela 2-5). A maioria dos UIMs que fazem uso de análise de modelos ou protótipos não está sendo completamente automatizado. No entanto, UIMs que fazem uso de análise de código HTML estão sendo parcialmente ou totalmente automatizadas. Analisadores HTML possuem regras integradas ao seu sistema que lhes permitem verificar se o código analisado constrói

aplicações *Web* que seguem os padrões de usabilidade. Como consequência, devido a que este tipo de UIMs encontra problemas de usabilidade relacionados basicamente com cores e padrões, eles são capazes de inspecionar a usabilidade sem a intervenção de inspetores. Existe uma carência de UIMs capazes de identificar problemas de usabilidade em modelos ou protótipos de aplicações *Web*. A análise dos artigos permitiu identificar que a maioria dos UIMs não consegue avaliar os fatores de usabilidade dependentes da aplicação sem a intervenção de um inspetor.

As descobertas relacionadas à avaliação de usabilidade de atributos genéricos e específicos de aplicações *Web* permitiram identificar que existe um baixo número de UIMs avaliando aplicações *Web* específicas. A categorização deste tipo de UIMs foca no contexto de uso, permitindo a avaliação de aspectos não considerados por UIMs genéricas. Por exemplo, alguns artigos (S05 e S24) apresentam UIMs que categorizam aplicações *Web* dentro do subconjunto que estão avaliando. Esta característica permite aos inspetores filtrar os fatores de usabilidade a fim de aprimorar a avaliação dos problemas de usabilidade. Consequentemente, existe uma necessidade de UIMs que avaliem aplicações *Web* específicas a fim de incrementar a precisão na avaliação de cada tipo de aplicação *Web*.

Segundo Fernandez *et al.* [12], a maioria dos UIMs de aplicações *Web* estão sendo empregados nas fases de elicitação de requisitos, design e implementação. Por meio da extensão do mapeamento sistemático foi descoberto que a maioria das novas propostas de UIMs focam nas fases de design e implementação. Isto pode ser observado na Tabela 2-5 já que a maioria dos UIMs basicamente faz uso de análise de modelos, código e/ou protótipos funcionais. A análise de modelos é aplicada na maioria dos casos durante a fase de elicitação de requisitos e design, enquanto a análise de código e protótipos é executada durante as fase de implementação. A maioria das análises de modelos foi executada quando todos os requisitos já tinham sido especificados. A correção de problemas de usabilidade nas últimas etapas do processo de desenvolvimento de software pode ser uma atividade custosa. Consequentemente, novas pesquisas devem ser orientadas a avaliar artefatos *Web* nas primeiras etapas do processo de desenvolvimento. Por exemplo, integrar requisitos de usabilidade durante a modelagem de um sistema e avaliar a usabilidade nas etapas iniciais do processo de desenvolvimento pode ajudar a diminuir os problemas encontrados em etapas posteriores. Por conseguinte, UIMs poderiam diminuir o número de problemas de usabilidade e incrementar a qualidade das aplicações *Web*.

A indústria de software deve considerar que nenhum dos UIMs é capaz de encontrar todos os problemas de usabilidade. No entanto, eles podem ser combinados para melhorar os resultados da avaliação e encontrar mais problemas de usabilidade. Durante a revisão foi encontrada apenas uma técnica (S18) que encontra problemas de usabilidade durante o processo de elicitação de requisitos. As outras técnicas podem ser aplicadas nas fases de design e implementação. Conseqüentemente, a indústria de software deve aplicar estas técnicas em diferentes fases do processo de desenvolvimento para aumentar a sua efetividade.

Para melhorar a eficiência muitas ferramentas têm sido propostas. Além disso, a indústria de desenvolvimento de software pode fazer uso de analisadores HTML para detectar problemas de usabilidade relacionados a cores e padrões. Contudo, estes analisadores não podem substituir inspetores, pois, estes conseguem encontrar problemas de usabilidade mais subjetivos. A indústria de desenvolvimento de software deve considerar que cada aplicação *Web* tem seus atributos específicos e, como consequência, as UIMs usadas para avaliá-las devem ser adaptadas para focar nas expectativas dos usuários para um determinado contexto de uso. Todos os atributos de usabilidade da *UIM* aplicada devem ser considerados a fim de executar avaliações mais consistentes e completas. Estas informações podem ser úteis para identificar problemas de usabilidade e incrementar os resultados da avaliação.

2.7 Considerações Finais

Diferentes métodos de inspeção têm sido propostos na última década para avaliar a usabilidade de aplicações *Web*. No entanto, a indústria de desenvolvimento de software não os está usando [15]. Um motivo para a não utilização destes métodos pode ser a falta de informação dos mesmos. Apesar dos esforços dos pesquisadores para resumir o conhecimento relacionado a *UEMs*, estas poucas revisões não descrever completamente o estado atual dos novos UIMs propostos. Existe a necessidade de identificar de forma organizada os UIMs de aplicações *Web* para atingir as necessidades da indústria.

Neste capítulo foi apresentada a extensão de um mapeamento sistemático sobre *UEMs* de aplicações *Web*, que descreve rigorosamente as novas propostas de UIMs usadas para examinar artefatos *Web*. Do conjunto inicial de artigos de Fernandez *et al.* [12] foram selecionados 26 estudos sobre UIMs. Foram criadas novas sub-questões de pesquisa para extrair dados dos estudos selecionados e categorizar os UIMs

encontrados. A análise dos resultados permitiu identificar o estado atual dos Métodos de Inspeção de Usabilidade de aplicações *Web*.

Os resultados demonstraram que os UIMs que estão sendo totalmente automatizados são os que fazem uso de análise de código HTML. Os UIMs que analisam modelos e/ou aplicações ou protótipos funcionais ainda dependem de inspetores e sua experiência para obter resultados satisfatórios. Os UIMs propostos estão fazendo uso de métodos base como a Avaliação Heurística, o Walkthrough Cognitivo, e o uso de perspectivas para executar inspeções. Além disso, alguns desses métodos focam em tipos de aplicações *Web* específicos.

Os UIMs propostos estão sendo aplicados nas últimas etapas do processo de desenvolvimento e, como consequência, o custo da redução dos problemas de usabilidade encontrados é alto. No entanto, novas propostas estão surgindo para incrementar a eficiência e eficácia na detecção de defeitos nas primeiras etapas do processo de desenvolvimento de aplicações *Web*.

Esta análise demonstrou que existem muitas soluções que podem ser aplicadas com respeito a UIMs. Contudo, ainda existem oportunidades de melhoria, particularmente na automatização dos métodos propostos e na inclusão dos mesmos nas primeiras etapas do processo de desenvolvimento de aplicações *Web*. Os dados obtidos através da extensão do mapeamento sistemático podem servir para promover e melhorar o atual uso e a pesquisa da usabilidade na *Web*. Além disso, os resultados fornecem informações sobre as opções de UIMs que existem e como aplicá-las para avaliar artefatos *Web*.

3 A Técnica *Web Design Usability Evaluation*

*Este capítulo apresenta as características que as novas propostas de métodos de inspeção de usabilidade devem possuir para apoiar a indústria de desenvolvimento de software. Estas características foram utilizadas para propor um novo método de inspeção de usabilidade: a técnica *Web Design Usability Evaluation* – *Web DUE*. A técnica *Web DUE* permite a inspeção de aplicações *Web* durante as primeiras etapas do processo de desenvolvimento e direciona os inspetores durante o processo de inspeção.*

3.1 Introdução

No Capítulo anterior, foi apresentada a extensão de um mapeamento sistemático sobre Métodos de Inspeção de Usabilidade (*Usability Inspection Methods* - *UIMs*) para aplicações *Web*. Nesta extensão, cada um dos artigos selecionados foi analisado de forma rigorosa com o objetivo de identificar o estado atual da arte no que diz respeito a *UIMs*, e disponibilizar os conceitos básicos para entender o seu uso. Estas informações podem auxiliar tanto os pesquisadores como os profissionais da área na escolha e utilização de um determinado *UIM* de aplicações *Web*.

Neste Capítulo, foram utilizados os resultados da extensão do mapeamento sistemático para sugerir uma série de características que *UIMs* emergentes devem possuir para apoiar a indústria de desenvolvimento de software. Com base nesses resultados, foi proposta a técnica de inspeção *Web Design Usability Evaluation* (*Web DUE*). Esta técnica tem por objetivo guiar os inspetores pelo processo de avaliação de protótipos em papel de aplicações *Web*, por meio da divisão de páginas *Web* em zonas típicas de páginas *Web*. Uma zona de página *Web* é uma parte da página que possui um tipo específico de conteúdo.

A técnica *Web DUE* foi baseada na técnica *Web Design Perspective* - *WDP* [6] que foi brevemente descrita no Capítulo anterior. Para criar a técnica *Web DUE*, nós relacionamos os atributos de usabilidade descritos nos itens de verificação da técnica

WDP com as zonas de páginas *Web* utilizadas na técnica *Web DUE*. Neste Capítulo, será apresentado o processo de criação da técnica, e um exemplo prático da sua aplicação para ilustrar o seu processo de inspeção.

Os resultados apresentados neste capítulo foram publicados no artigo: *Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications* [31], publicado no *International Conference on Software Engineering & Knowledge Engineering (SEKE) 2012*.

Este capítulo está organizado da seguinte forma: a Seção 3.2 apresenta o conjunto de características que novos UIMs devem possuir para apoiar a indústria de desenvolvimento de software, e os relaciona com a extensão do mapeamento sistemático apresentado no capítulo anterior. Na Seção 3.3 é apresentada a proposta inicial da técnica *Web DUE* descrevendo os conceitos relacionados à sua concepção. A Seção 3.4 apresenta o processo de inspeção da técnica *Web DUE* e uma prova de conceito onde são inspecionados os protótipos em papel de uma aplicação *Web* real. Finalmente a Seção 3.5 apresenta as considerações finais deste capítulo.

3.2 Apoiando a Indústria de Desenvolvimento de Software

Na Tabela 3-1 são apresentados os resultados resumidos da extensão do mapeamento sistemático sobre métodos de inspeção de usabilidade de aplicações *Web*. Foram utilizadas as respostas a estas sub-questões de pesquisa para identificar quais características deviam ser consideradas durante a concepção dos novos métodos de inspeção de usabilidade. Estas características visam apoiar a indústria de desenvolvimento de software no que diz respeito à garantia da qualidade de aplicações *Web* por meio de métodos de inspeção de usabilidade.

Como pode ser observado, a sub-questão Q1 (Método Base) tem caráter exploratório, visando identificar quais teorias estão sendo utilizadas para a concepção de novos UIMs. Consequentemente, esta questão não foi considerada para a elaboração das características que visam identificar oportunidades de pesquisa. A seguir são mostradas as características propostas e o seu relacionamento com os resultados das sub-questões de pesquisa.

Tabela 3-1 Resumo dos resultados do mapeamento sistemático apresentados no Capítulo 2.

Q1. Método Base			
Avaliação Heurística	Walkthrough Cognitivo	Baseado em Perspectivas	Outra Base
26,92%	19,23%	15,38%	57,69%
Q2. Tipo de Artefato Avaliado			
Análise de Código HTML	Análise de Modelo	Análise de Aplicação/Protótipo	
15,38%	15,38%	76,92%	
Q3. Tipo de Aplicação Web avaliada pelo Método de Inspeção de Usabilidade			
Genérico		Específico	
88,46%		11,54%	
Q4. Uso de Inspectores no Processo de Inspeção			
Precisa da Intervenção de Inspectores		Totalmente Automatizado	
92,31%		7,69%	

Característica #1: Habilidade para encontrar problemas de usabilidade durante as primeiras etapas do processo de desenvolvimento: Os resultados para a sub-questão Q2 indicaram que as novas propostas de UIMs devem conseguir identificar problemas de usabilidade durante as etapas de design e análise. Esta característica visa diminuir os custos de correção de problemas de usabilidade visto que o código fonte do sistema ainda não foi desenvolvido. Consequentemente, as novas propostas de métodos de inspeção de usabilidade devem ser capazes de avaliar modelos e protótipos.

Característica #2: Habilidade de encontrar problemas específicos de aplicações Web: A sub-questão de pesquisa Q3 indicou que a maioria dos métodos de inspeção genéricos não oferecem *feedback* sobre como tratar um problema de usabilidade uma vez o mesmo é encontrado. Consequentemente, os UIMs emergentes devem ser capazes de apoiar tanto na identificação como na solução dos problemas de usabilidade.

Característica #3: Automação: Os resultados para a sub-questão de pesquisa Q4 indicam que existe a necessidade de apoio ferramental para as novas abordagens de inspeção de usabilidade. Como consequência, para conseguir a melhoria do desempenho da avaliação, os novos UIMs devem ser automatizados ou oferecer apoio ferramental.

Esta característica deve ser planejada com cuidado, pois nem todo UIM pode ser automatizado. Nesse sentido, é necessário providenciar meios para reduzir o esforço do inspetor ao executar a avaliação de usabilidade.

3.3 A Concepção da Técnica *Web Due*

A técnica *Web Design Usability Evaluation (Web DUE)* é uma técnica que tem por objetivo apoiar a indústria de desenvolvimento de software no que diz respeito a inspeções de usabilidade de aplicações *Web*. Para atingir esse objetivo, a técnica *Web DUE* foi criada seguindo as características decorrentes da análise dos resultados da extensão em cima do mapeamento sistemático em [12], propostas na Seção 3.2. Nesta seção, serão descritas detalhadamente cada uma das características da técnica *Web DUE* e os conceitos por trás da sua concepção.

3.3.1 Avaliação da Usabilidade de Protótipos de Baixa Fidelidade (ou Mockups)

Uma das principais características da técnica *Web DUE* é sua habilidade de encontrar problemas de usabilidade durante as primeiras etapas do processo de desenvolvimento. Como sugerido na Característica #1, a técnica *Web DUE* pode ser utilizada para avaliar a usabilidade de protótipos em papel ou *mockups*. Um *mockup* é uma imagem de como seria a interface do usuário se esta fosse implementada. A ideia de utilizar mockups veio do custo necessário para seu desenvolvimento. Uma vez que um mockup pode ser desenhado em papel usando um lápis, o seu custo é baixo. Isto torna o mockup um modelo de baixo custo e alta eficiência e eficácia para a validação do sistema com o usuário e para a avaliação de usabilidade. O uso de mockups para realizar a inspeção é importante visto que encontrar problemas de usabilidade no início do desenvolvimento pode reduzir o custo de correção dos mesmos [38].

3.3.2 Processo de Inspeção Guiado por Zonas Próprias de Páginas *Web*

A maior diferencial da técnica *Web DUE* é sua habilidade de guiar os inspetores pelo processo de avaliação, por meio de “peças” específicas que são usadas para compor páginas *Web* de aplicações *Web*. Estas “peças” são denominadas de Zonas de Páginas *Web* e contém componentes específicos de aplicações *Web* [13]. A ideia de utilizar zonas para guiar os inspetores veio da leitura de artigos relacionados com a classificação de páginas e aplicações *Web*. Optou-se por usar zonas de páginas *Web*

para guiar os inspetores, visto que estas são de fácil entendimento e podem ser usadas para compor qualquer tipo de página [13]. A Tabela 3-2 apresenta a lista das zonas de páginas *Web* utilizada para criar a técnica *Web DUE*. Esta lista foi baseada na lista proposta por Fons *et al.* [13]. No entanto, os autores não consideraram a criação de uma zona de Ajuda. Uma zona de ajuda pode ser aquela que possibilite ao usuário obter informações relacionadas ao uso da aplicação e como funciona. Além disso, Nielsen [26] afirma que prover ajuda aos usuários é um dos atributos de usabilidade mais importante. Consequentemente, a zona de Ajuda foi adicionada ao conjunto inicial proposto em [13] para direcionar os inspetores a avaliar esse atributo. Para descrever detalhadamente cada zona, a Tabela 3-2 também apresenta a adequação de incluir de cada zona em uma página *Web*: (a) Obrigatório: indica que a zona *Web* sempre deve estar presente na página *Web*; (b) Opcional: indica que a zona *Web* pode estar presente na Página *Web*, mas a sua ausência não afetará a funcionalidade da página; e (c) Depende da Funcionalidade: indica que dependendo do propósito da página, os designers devem ou não incluir a zona *Web*.

Tabela 3-2 Lista de zonas de páginas *Web* baseadas em [13], conteúdo e adequação para sua inclusão em uma página *Web*.

Zona de Página <i>Web</i>	Adequação	Conteúdo
Navegação	Obrigatória	Links de Navegação
Estado do Sistema	Obrigatória	Informação sobre o estado da aplicação e como se chegou ai.
Informação	Depende da Funcionalidade	Informações provenientes do Banco de Dados da Aplicação.
Serviços	Depende da Funcionalidade	Acesso a funcionalidades relacionadas com a zona de informação.
Informação do Usuário	Depende da Funcionalidade	Informação sobre o usuário logado.
Acesso Direto	Depende da Funcionalidade	Links para funcionalidade <i>Web</i> comuns na aplicação (Login, Logout, Home, etc.)
Entrada de Dados	Depende da Funcionalidade	Oferece uma opção de entrada de dados para executar operações no sistema.
Instituição	Opcional	Informação sobre a instituição responsável pela aplicação <i>Web</i> .
Customização	Opcional	Conteúdo independente do domínio da aplicação.

Zona de Página Web	Adequação	Conteúdo
Ajuda	Obrigatória	Informação sobre como usar a aplicação e como funciona.

A maior vantagem do uso de Zonas de Páginas Web é que estas podem direcionar o inspetor a avaliar apenas os elementos presentes na aplicação. Em outras palavras, os inspetores só deverão verificar os atributos de usabilidade para aquelas zonas que compõem o *mockup* avaliado. Outro benefício do uso de zonas, é que elas podem ajudar os inspetores a identificar os elementos dentro da aplicação, permitindo focar em atributos de usabilidade relacionados a esses elementos durante a execução da inspeção.

Além do uso de zonas de páginas Web, a técnica *Web DUE* foi baseada na técnica *Web Design Perspective Based* (WDP) [6]. A técnica WDP, como descrito na Seção 2.5, relaciona a usabilidade das aplicações Web a três principais perspectivas: conceituação, navegação e apresentação. Ao usar estas perspectivas e as Heurísticas de Nielsen [26], a técnica WDP cria pares Heurística×Perspectiva (H×P) ao combinar as heurísticas e perspectivas quando existe um relacionamento entre elas. Para cada par H×P, a técnica WDP oferece itens de verificação para apoiar os inspetores durante o processo de identificação de problemas de usabilidade de aplicações Web. A Tabela 3-3 apresenta os relacionamentos entre as heurísticas e perspectivas da técnica WDP, enquanto a Figura 3-1 apresenta parte dos itens de verificação da técnica WDP.

Tabela 3-3 Descrição das perspectivas da técnica WDP [6] e seus respectivos relacionamentos com as heurísticas de Nielsen [26].

Heurísticas de Nielsen	Perspectiva		
	Apresentação	Conceituação	Navegação
Visibilidade do estado do sistema	A1	C1	
Concordância entre o sistema e o mundo real	A2	C2	
Controle e liberdade ao usuário			N3
Consistência e padrões	A4	C4	
Prevenção de erros	A5		N5
Reconhecer ao invés de lembrar	A6	C6	
Flexibilidade e eficiência de uso	A7		N7

Heurísticas de Nielsen	Perspectiva		
	Apresentação	Conceituação	Navegação
Projeto minimalista e estético	A8		
Reconhecimento, diagnóstico e recuperação de erros	A9	C9	N9
Ajuda e documentação	A10	C10	N10
Descrição das Perspectivas			
Conceituação: Representa os elementos conceituais que fazem parte do domínio da aplicação.			
Apresentação: Representa as características relacionadas ao layout e distribuição dos elementos na interface.			
Navegação: Representa o espaço navegacional, definindo os elementos de acesso a informação e sua associação.			

A.1. Visibilidade do estado do sistema

- Avalie se o estado do sistema está sempre visível para o usuário.
 - Estado do sistema é o local da aplicação onde o usuário se encontra ou o estado após uma transação.

A.2. Concordância entre o sistema e o mundo real

- Avalie se a informação e as opções do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.

Figura 3-1 Extrato da técnica WDP [6] contendo parte dos itens de verificação.

A técnica *Web DUE* também possui descrições sobre como identificar problemas de usabilidade. Foram extraídos os itens de verificação da técnica WDP e relacionados com as zonas próprias de páginas *Web*. Consequentemente, foi gerado um *checklist* de itens de verificação para cada uma das zonas de páginas *Web*. Os itens de cada *checklist* servem para checar as propriedades de usabilidade de cada zona. Além disso, de forma a apoiar o processo de inspeção, agrupamos cada um dos itens de verificação com um exemplo/esclarecimento. Os inspetores que tenham dúvidas sobre um determinado item podem-se referir a estes exemplos para obter maiores informações sobre como identificar e avaliar um problema de usabilidade. A Tabela 3-4 apresenta parte do *checklist* de verificação da zona de Entrada de Dados. Nessa tabela é apresentada a descrição da zona, assim como quatro itens de verificação. O conjunto completo de *checklists* pode ser verificado no Apêndice B.

Tabela 3-4 Parte do *checklist* de itens de verificação da zona de entrada de dados.

Zona de Entrada de Dados	
<p>Descrição: Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.</p>	
Itens de Verificação	Exemplos/Esclarecimentos
É fácil ver (encontrar, localizar) a zona de entrada de dados no sistema.	O usuário consegue ver facilmente onde é que se inserem dados no sistema.
É fácil entender que é nesta zona que se inserem dados.	Os campos de entrada de dados (Combo Box, Line Edit, Dial, Horizontal Slider, etc.) devem ser facilmente reconhecíveis. O usuário deve identificar que eles servem para informar dados.
A interface indica que dados devem ser preenchidos de forma obrigatória.	Por exemplo, o sistema informa com um “*” ou um “obrigatório” os campos obrigatórios.
A interface indica qual é o formato correto de entrada dos dados a serem inseridos.	Os campos de entrada de dados devem prover dicas sobre como preenchê-los. Por exemplo, um dado relacionado com Data, pode conter a seguinte dica: “dd/mm/aa”.

A característica #2, da Seção 3.2, indica que novas UIMs de aplicações *Web* devem apoiar os inspetores tanto na identificação quanto na correção dos problemas de usabilidade. A técnica *Web DUE* consegue direcionar os inspetores a avaliar apenas os itens relevantes na aplicação garantindo a identificação dos problemas de usabilidade relacionados com a aplicação inspecionada. Além disso, uma vez que o inspetor identifica o item violado na interface, a técnica *Web DUE* deixa claro que para que o problema seja consertado, a interface deve estar aderente aos itens de verificação. Como consequência, a técnica *Web DUE* também informa o que deve ser melhorado para aumentar a usabilidade dos sistemas *Web*.

3.3.3 Uso de Apoio Ferramental para reduzir o Esforço dos Inspectores

Outra das características importantes da técnica *Web DUE* é que conta com apoio ferramental para reduzir o esforço dos inspetores durante o processo de inspeção (ver Característica #3 da Seção 3.2). Uma vez que a técnica envolve o julgamento dos inspetores para realizar uma avaliação de usabilidade, a *Web DUE* não é totalmente automatizada. No entanto, ela conta com o auxílio da ferramenta *Mockup Design Usability Evaluation (Mockup DUE)*. A *Mockup DUE* surgiu a partir dos resultados do

estudo de viabilidade da técnica *Web DUE*. A ferramenta apresenta de forma automática os itens de verificação da técnica, enquanto realiza o mapeamento dos *mockups*, simulando a interação entre o sistema e o usuário. No Capítulo 5 será feito um detalhamento das características da ferramenta *Mockup DUE*.

3.4 Aplicando a Técnica *Web DUE* na Inspeção de *Mockups*

A técnica *Web DUE* possui uma série de passos a serem seguidos de forma a realizar uma inspeção de usabilidade sobre os *mockups* de uma aplicação *Web*. A seguir, serão apresentadas cada uma destas etapas e, para exemplificar o seu uso, será apresentada uma prova de conceito na que os *mockups* de uma aplicação *Web* serão inspecionados.

3.4.1 O Processo de Inspeção da Técnica *Web DUE*

A Figura 3-2 apresenta o processo de inspeção da técnica *Web DUE* contendo quatro etapas: (a) preparar *mockups* (etapa prévia ao processo de detecção de defeitos), (b) dividir *mockups* em zonas, (c) checar os itens de verificação, e (d) identificar os problemas de usabilidade. A primeira etapa, preparar *mockups*, consiste no mapeamento da interação que o usuário terá com os *mockups*. Isto é, como a interface se comportará ao realizar ações sobre ela. Para isso, o designer da aplicação deverá informar que componentes dentro da aplicação serão usados pelo usuário para interagir. É com base nas ações do usuário sobre esses componentes que os *mockups* deverão ser apresentados. Por exemplo, na Figura 3-2 o mapeamento indica que o *mockup* 1 poderá levar aos *mockups* 2 ou 3 dependendo das ações do usuário.

Na etapa 2 da Figura 3-2, é apresentada a etapa na qual o inspetor deverá dividir os *mockups* a serem inspecionados de acordo com as zonas próprias de páginas *Web*. Nesta etapa o inspetor deverá identificar, para cada *mockup*, quais são as zonas que o mesmo possui. Esta etapa é crucial, pois, é com base nessa escolha que o inspetor irá realizar a avaliação de usabilidade.

As etapas 3 e 4 da Figura 3-2 são realizadas em paralelo. Para cada zona de páginas *Web* identificada, o inspetor deverá verificar o *checklist* correspondente (Etapa 3 da Figura 3-2). Caso aconteça uma não conformidade em relação a algum dos itens de verificação, o inspetor deverá marcar essa não conformidade como um problema de usabilidade. Para apontá-lo, o inspetor pode riscar o *mockup* apontando aonde é que o problema aconteceu (Etapa 4 da Figura 3-2). Vale resaltar que um *mockup* pode não

conter todas as zonas ou pode ter várias zonas repetidas, porém espalhadas. No caso de não conter uma zona, o inspetor deverá pular esse *checklist* e verificar apenas os *checklists* das zonas contidas. No caso de ter a mesma zona repetida, porém espalhada dentro do mesmo *mockup*, o inspetor pode verificar todas as zonas repetidas com o mesmo *checklist*. O processo de inspeção pode ser executado de formas diferentes: (a) avaliar um *mockup* de cada vez com vários *checklists* segundo as zonas encontradas no *mockup*, ou (b) utilizar um mesmo *checklist* para avaliar a mesma zona em todos os *mockups* onde esta aparece. Cabe ao inspetor decidir qual é a melhor forma de realizar a inspeção, porém sempre apontando os defeitos no próprio *mockup* como sugerido na etapa 4 da Figura 3-2.

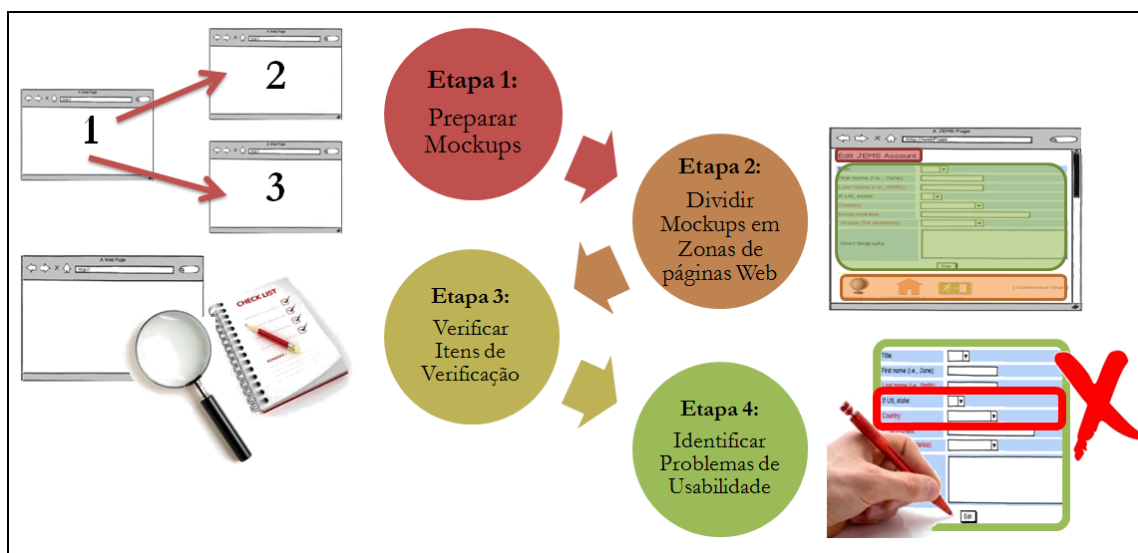


Figura 3-2 Processo de inspeção da técnica *Web DUE*.

Vale ressaltar que para facilitar o seguimento destas etapas, a técnica *Web DUE* conta com um conjunto de instruções que indicam os passos a serem seguidos durante o processo de inspeção. Estas instruções também são apresentadas no Apêndice B.

3.4.2 Prova de Conceito da Técnica *Web DUE*

Nesta seção, é apresentada uma prova de conceito ao avaliar a usabilidade de um protótipo em papel utilizando a técnica *Web DUE*. Esta avaliação foi feita para exemplificar as etapas do processo de inspeção da técnica e, por tanto, não foi feita uma avaliação exaustiva de todos os *mockups* da aplicação *Web* e sim de um subconjunto destes. Os *mockups* utilizados nesta prova de conceito foram baseados nas páginas *Web*

de uma aplicação real. A aplicação *Web* utilizada como base foi o portal do MPS-Br¹ (Melhoria de Processo de Software Brasileiro). Além disso, nenhum problema de usabilidade foi inserido nos *mockups*, para garantir que os problemas identificados fossem reais. As Figuras 3-3 e 3-4 apresentam os *mockups* baseados no portal do MPS.Br. Estes *mockups* apresentam a página de guias e a página de visualização de um guia respectivamente.

Como sugerido na Seção 3.3, a primeira etapa do processo de inspeção, consiste no mapeamento dos *mockups*. Neste caso, o *Mockup* #1 e o *Mockup* #2 estão ligados através do link “Guia Geral de Serviços: 2012 (Agosto de 2012)”. Ou seja, quando o usuário clicar no link estando no *mockup* #1, o sistema carregará o *mockup* #2. Vale ressaltar que podem existir mais elementos de interação no *mockup*, no entanto nesta etapa o foco é a identificação de elementos que representem uma transição de um *mockup* para outro. Como consequência, só foram relatados os elementos de transição.

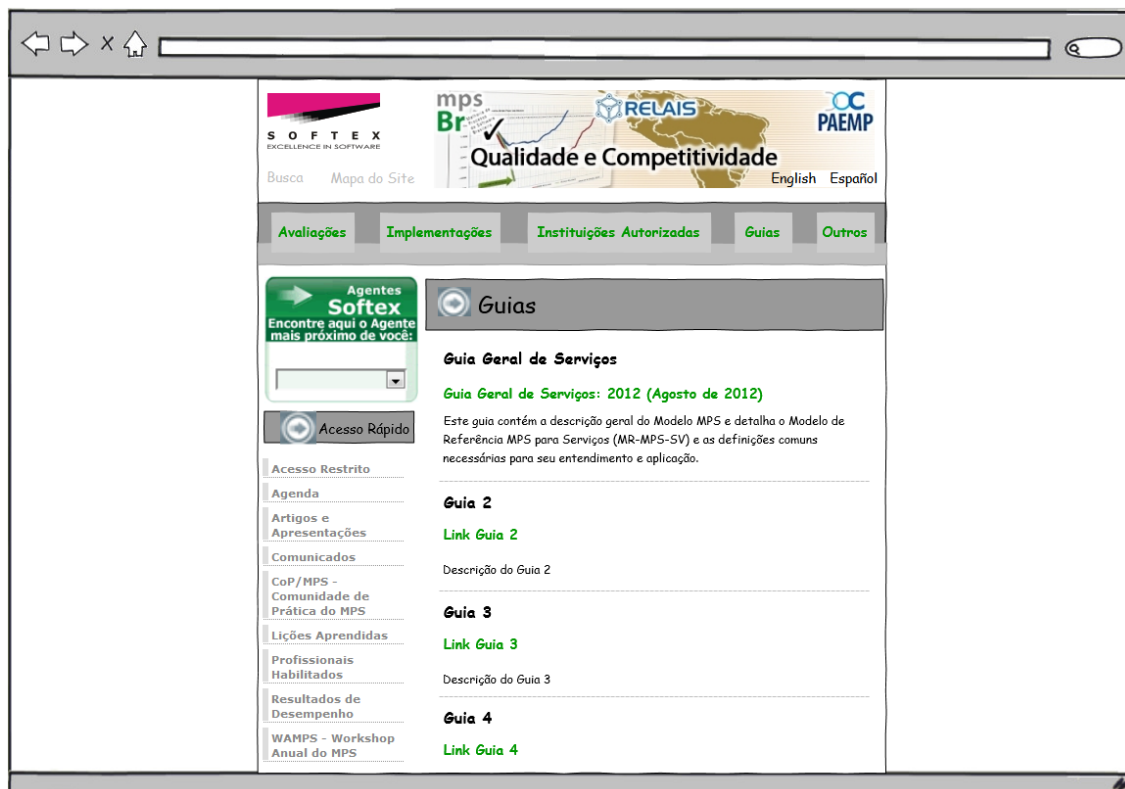


Figura 3-3 Mockup #1 do portal do MPS.Br: página de listagem de guias.

A segunda etapa consiste na identificação das zonas próprias de páginas *Web*. Nesta etapa, o inspetor deve identificar no *mockup* quais zonas estão presentes. A

¹ http://www.softex.br/mpsbr/_home/default.asp#

Figura 3-5 mostra a divisão dos *mockups* do portal do MPS.Br. Cada uma das zonas identificadas está marcada com uma cor diferente. Caso uma zona se repita, a mesma cor será usada para identificá-la. Além disso, cada uma das zonas encontradas foi numerada como segue: (1) Zona da Instituição, que apresenta o logo do MPS.Br; (2) Zona de Serviços, que apresenta links relacionados com os dados do sistema; (3) Zona de Navegação, que apresenta o menu que usuário pode usar para acessar as diferentes funcionalidades ou informações do sistema; (4) Zona do Estado do Sistema, que informa o usuário em que parte do sistema ele se encontra; (5) Zona de Informação, que contém os dados do banco de dados do sistema; e (6) Zona de Acesso Direto, que contém atalhos para as diferentes funcionalidades do sistema. Deve-se notar que as zonas podem-se repetir dentro do mesmo *mockup* e entre os *mockups*. Além disso, as zonas podem se sobrepor, ou seja, uma determinada parte do *mockup* pode se encaixar na descrição de mais de uma zona. Nesses casos, cabe ao inspetor julgar qual das zonas é a mais adequada para categorizar as “peças” da interface.



Figura 3-4 Mockup #2 do portal do MPS.Br: página apresentando um guia.

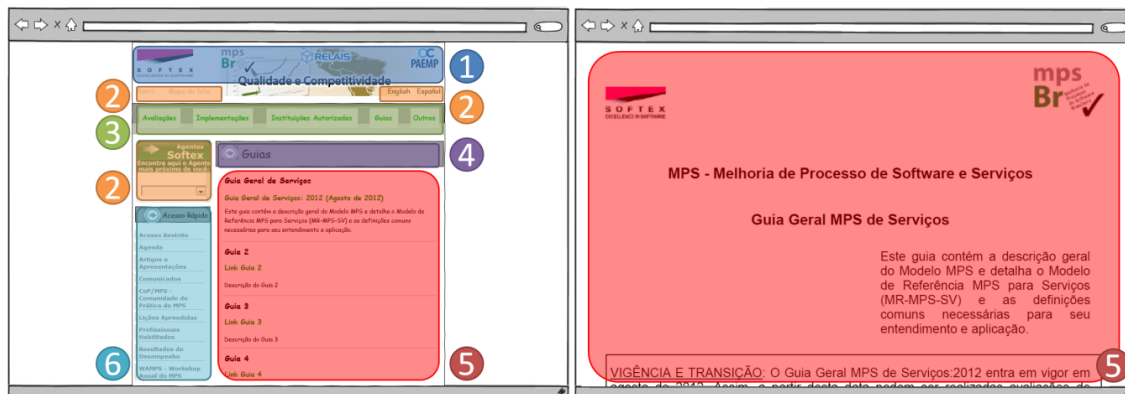


Figura 3-5 Divisão dos *mockups* em zonas próprias de páginas *Web*.

Para identificar problemas de usabilidade, os inspetores devem apontar que componentes da aplicação dentro de cada uma das zonas de páginas *Web* que não estão aderentes com os itens de verificação dos *checklists*. Na Tabela 3-5 são apresentados itens de verificação que foram violados nos *mockups*. Cada um destes itens de verificação está associado a uma zona de página *Web* e ao *mockup* que contem a zona. Na Figura 3-6 foram aumentados os elementos nos que os problemas de usabilidade foram identificados. Estes elementos serão marcados nos *mockups* como forma de apontar os defeitos. Estes passos condizem com as etapas 3 e 4 do processo de inspeção da técnica *Web DUE* (ver Seção 3.3). Nos seguintes parágrafos serão descritos os problemas de usabilidade identificados, associando os itens de verificação da Tabela 3-5 com os elementos aumentados da Figura 3-6.

Com relação ao *mockup* #1, foram apontados seis problemas. O item de verificação número 1 está relacionado ao elemento A da zona de Instituição do *mockup* # 1. O problema de usabilidade é que o significado da sigla MPS.Br não pode ser entendido pois as letras estão ilegíveis. Como consequência, um usuário que não souber o significado da sigla MPS.Br não terá informações sobre a mesma. Outro problema similar pode ser percebido ao verificar o item violado número 2 que está relacionado ao elemento B na zona de serviços. Novamente não é possível visualizar a informação, pois as letras são muito claras, diminuindo o contraste. Por sua vez, o item violado 3, relacionado ao elemento C, aponta um problema de usabilidade. O menu não é padronizado pois o tamanho de cada botão varia. Esse tipo de descuido com o design da aplicação pode trazer desconforto para o usuário. Outro problema apontado é o tipo de representação utilizado no estado do sistema (Item violado número 4 relacionado ao elemento D). É estranho utilizar uma seta como ícone para o estado do sistema. Na Tabela 3-5 também aparece o item violado 5 na zona de acesso direto. Ao avaliar o

elemento E foi percebido que o sistema tem uma área de acesso restrito, onde o usuário pode acessar informações específicas. No entanto, não existe nenhuma zona de informação do usuário, seja indicando que o usuário fez o login, ou uma vez tendo feito o login, informações relacionadas a ele, como identificadores (nome, e-mail, etc.). Nesse sentido, quando uma zona se torna obrigatória na página e esta não aparece, existe um problema de usabilidade. Finalmente, o item violado número 6 aponta um problema na zona de informação do *mockup* #1. O elemento F mostra como o sistema sobrecarrega o usuário ao repetir informações.

Com relação ao *mockup* #2 só foram apontados dois itens violados. Como descrito anteriormente, se uma zona de páginas *Web* se torna obrigatória, seja por causa das funcionalidades da página, ou princípios de usabilidade, o sistema deve apresentar a zona ao usuário. Nesse sentido, apesar de ser uma página apresentando um guia, a página deve ainda apresentar as zonas de navegação e de ajuda (consideramos que a zona de estado do sistema é o título do guia). Portanto, os itens de verificação 7 e 8 foram violados, o que representa um problema de usabilidade.

Tabela 3-5 Itens de verificação violados pelos *mockups* do portal do MPS.Br.

Número	<i>Mockup</i>	Zona de Páginas <i>Web</i>	Item de Verificação Violado
1	<i>Mockup</i> 1	Instituição	A informação é legível.
2	<i>Mockup</i> 1	Serviços	A informação é legível.
3	<i>Mockup</i> 1	Navegação	O design é agradável.
4	<i>Mockup</i> 1	Estado do Sistema	O estado do sistema está expresso em uma representação facilmente compreendida pelo usuário.
5	<i>Mockup</i> 1	Informação do Usuário	Existe uma zona de Informação de Usuário
6	<i>Mockup</i> 1	Informação	A informação é útil.
7	<i>Mockup</i> 2	Navegação	Existe uma zona de navegação no sistema
8	<i>Mockup</i> 2	Ajuda	Existe uma zona de ajuda no sistema

Esta seção apresentou um exemplo prático de aplicação da técnica *Web DUE* para identificar problemas de usabilidade de *mockups* de aplicações *Web*. Como mencionado anteriormente, neste exemplo não foi feita uma inspeção exaustiva, e sim

uma inspeção de alguns elementos do *mockup* para exemplificar o uso da técnica. Conseqüentemente, para realizar uma avaliação completa, todos os *mockups* da aplicação *Web* deverão ser avaliados. Além disso, para corrigir os problemas encontrados, os *mockups* devem integrar os itens de verificação apontados na Tabela 3-5. Na próxima seção serão apresentadas as conclusões deste capítulo.



Figura 3-6 Elementos da interface associados aos itens de verificação violados.

3.5 Considerações Finais

Este Capítulo discutiu os resultados da extensão do mapeamento sistemático do Capítulo 2. Essa análise teve por objetivo propor uma série de características a fim de apoiar a indústria de desenvolvimento de software com respeito a inspeções de usabilidade de aplicações *Web*. Como resultado, foi possível identificar três características: (a) encontrar problemas nas primeiras etapas do processo de desenvolvimento, (b) encontrar problemas específicos e sugerir melhorias para corrigi-los, e (c) automatizar o processo de inspeção ou oferecer apoio ferramental para diminuir o esforço da inspeção.

Estas características foram utilizadas para propor a técnica de inspeção *Web DUE* que guia os inspetores pelo processo de inspeção usando zonas próprias de páginas *Web*. A *Web DUE* também permite avaliar os atributos nas primeiras etapas do processo de desenvolvimento ao avaliar protótipos de baixa fidelidade ou *mockups*. Neste capítulo também foi apresentada uma prova de conceito do processo de inspeção da técnica.

Esses resultados podem servir como base para a criação de novas técnicas de inspeção de usabilidade de aplicações *Web*. As novas propostas de métodos de inspeção de usabilidade podem seguir estas sugestões para apoiar a indústria de desenvolvimento de software. Nos próximos capítulos serão apresentados o estudo de viabilidade da técnica, e os resultados que levaram à concepção da ferramenta de apoio *Mockup DUE*.

4 O Estudo de Viabilidade da Técnica *Web DUE*

*Este capítulo apresenta um estudo experimental executado para verificar se os resultados da técnica são viáveis e se o tempo é bem empregado. Nesse estudo, foi feito um comparativo entre o desempenho da técnica *Web DUE* e a sua técnica base, a técnica *WDP*. A análise dos resultados quantitativos e qualitativos deste estudo oferece informações sobre a eficiência e eficácia da técnica *Web DUE*, assim como dados relevantes do que pode ser modificado para melhorar seu desempenho.*

4.1 Introdução

No capítulo anterior, foi apresentada a técnica *Web Design Usability Evaluation (Web DUE)*. A técnica tem por objetivo reduzir o custo de correção de problemas de usabilidade de aplicações *Web* por meio da realização de inspeções de usabilidade durante as primeiras etapas do processo de desenvolvimento. Executar inspeções nessas etapas pode reduzir os custos de correção dos problemas, pois o código fonte da aplicação não foi escrito [38]. Nesse sentido, optou-se pela inspeção de *mockups*, que são protótipos de baixa fidelidade que mostram, através de imagens, a aparência da aplicação *Web* ao concluir o seu desenvolvimento.

Neste capítulo será apresentado um estudo experimental que foi executado com o propósito de verificar a viabilidade da técnica. Este estudo mediu os indicadores de eficácia e eficiência da técnica *Web DUE* e os comparou com os da técnica base (*WDP*). Ao comparar estes indicadores foi possível corroborar que a técnica *Web DUE* permite a identificação de mais problemas de usabilidade de *mockups* de aplicações *Web* em um tempo razoável. Além disso, os participantes do estudo responderam a um questionário com o objetivo de obter dados qualitativos sobre as dificuldades ao aplicar a técnica para executar uma inspeção de usabilidade. Estes resultados qualitativos, em conjunto com os resultados quantitativos, proporcionaram informações relevantes para a identificação de oportunidades de melhoria na técnica *Web DUE*.

Os resultados apresentados neste capítulo foram publicados no artigo: *Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications* [30], publicado no Simpósio Brasileiro de Engenharia de Software (SBES) 2012. Este artigo recebeu o prêmio de “Primeiro Melhor Trabalho” e foi chamado para extensão no *Journal of Software Engineering Research and Development*.

O restante deste capítulo foi dividido como segue. Na Seção 4.2 são apresentadas as etapas do estudo de viabilidade. A Seção 4.3 apresenta os resultados quantitativos e qualitativos, assim como a análise que permitiu sugerir modificações na técnica para melhorar seu desempenho. Na Seção 4.4, por sua vez, são apresentadas as ameaças à validade deste estudo experimental. Finalmente, na Seção 4.5, são apresentadas as considerações finais.

4.2 Estudo de Viabilidade

Segundo Shull *et al.* [35], o primeiro estudo que deve ser executado ao avaliar uma nova tecnologia é um estudo de viabilidade. Neste capítulo será descrito o estudo experimental que responde à seguinte questão de pesquisa: “A técnica *Web DUE* é viável em termos do número de defeitos de usabilidade encontrados, e é o tempo bem empregado?” Para responder a esta questão, foi feito um comparativo entre a técnica *Web DUE* e a técnica WDP [6]. Vale ressaltar que a técnica WDP não é própria para a inspeção de *mockups* de aplicações *Web*. No entanto, a técnica WDP é a técnica base da técnica *Web DUE* e, por tanto, é razoável verificar se a técnica *Web DUE* apresenta melhores resultados que a técnica WDP ao inspecionar *mockups* de aplicações *Web*.

Nas seguintes subseções serão apresentadas as etapas deste estudo experimental, providenciando informações para avaliar a usa qualidade e a aplicabilidade dos seus resultados.

4.2.1 Objetivo

A Tabela 4-1 apresenta o objetivo deste estudo de viabilidade usando o paradigma GQM (*Goal Question Metric*) [3]. Neste estudo, a viabilidade da técnica *Web DUE* foi caracterizada com base nos seus indicadores de eficácia e eficiência. Para estes indicadores, foram utilizadas as mesmas definições usadas em [6] e [11]:

Eficácia é a razão entre o número de problemas detectado pelo inspetor e o número de problemas existentes.

Eficiência é a razão entre o número de problemas detectado pelo inspetor e o tempo gasto por ele/ela para encontrá-los.

Tabela 4-1 Objetivo do estudo de viabilidade da técnica *Web DUE*.

Analisar	A técnica de inspeção <i>Web Design Usability Evaluation</i>
Com o propósito de	Caracterizá-la
Com respeito a	Sua viabilidade em termos dos indicadores de eficácia e eficiência ao ser comparada com a técnica WDP.
Do ponto de vista de	Pesquisadores em Engenharia de Software
No contexto de	Uma avaliação de usabilidade de <i>mockups</i> de uma aplicação <i>Web</i> executada por inspetores de usabilidade.

4.2.2 Hipóteses

Com base nos indicadores definidos acima, o estudo experimental foi planejado e conduzido para testar as seguintes hipóteses (nula e alternativa, respectivamente):

H₀₁: Não existe diferença, em termos de eficácia, ao utilizar a técnica *Web DUE* para encontrar problemas de usabilidades em *mockups* de aplicações *Web*, quando comparada com a técnica WDP.

H_{A1}: A técnica *Web DUE* apresenta diferentes resultados, em termos do indicador de eficácia, quando comparada com a técnica WDP.

H₀₂: Não existe diferença, em termos de eficiência, ao utilizar a técnica *Web DUE* para encontrar problemas de usabilidades em *mockups* de aplicações *Web*, quando comparada com a técnica WDP.

H_{A2}: A técnica *Web DUE* apresenta diferentes resultados, em termos do indicador de eficiência, quando comparada com a técnica WDP.

4.2.3 Participantes

O estudo foi conduzido em abril de 2012 com alunos da Universidade Federal do Amazonas do curso de Ciência da Computação. No total, oito alunos de graduação e pós-graduação participaram do estudo. Os participantes assinaram um Termo de Consentimento Livre e Esclarecido (TCLE) e preencheram um formulário de caracterização. O TCLE serviu para informar os participantes que estariam fazendo

parte de um estudo relacionado com inspeções de usabilidade de aplicações *Web*. O formulário de caracterização, por sua vez tinha por objetivo identificar a experiência dos participantes com relação a: (a) conhecimento em usabilidade, (b) experiência em avaliações de usabilidade, e (c) experiência no processo de análise e design de software. Todos os participantes responderam questões objetivas relacionadas ao seu grau de conhecimento e experiência profissional. Os dados da caracterização foram analisados e os participantes foram classificados como tendo Baixa (B), Média (M) ou Alta (A) experiência segundo com as informações disponibilizadas no seu formulário.

Para classificar a experiência dos inspetores no que diz respeito à interação humano computador (IHC), foi considerado: (a) Baixa, se o participante não tinha tido experiência prática em IHC e/ou não tinha estudado IHC; (b) Média, se o participante tinha estudado IHC, mas tinha pouca experiência prática em avaliação de usabilidade no geral; e (c) Alta, se o participante tinha estudado IHC em livros e em sala de aula, e tinha participado em projetos envolvendo avaliações de usabilidade.

Para classificar a experiência dos inspetores no que diz respeito a avaliações de usabilidade, foi considerado: (a) Baixa, se o participante não tinha tido experiência prática e/ou não tinha estudado sobre avaliações de usabilidade; (b) Média, se o participante tinha estudado alguma técnica de avaliação de usabilidade, mas tinha pouca experiência prática; e (c) Alta, se o participante tinha estudado mais de uma técnica de avaliação, e tinha participado em mais de duas avaliações de usabilidade.

Para classificar a experiência dos inspetores no que diz respeito à análise e design, foi considerado: (a) Baixa, se o participante não tinha tido experiência prática e/ou não tinha estudado sobre análise e design; (b) Média, se o participante tinha estudado análise ou design, mas tinha pouca experiência prática; e (c) Alta, se o participante tinha estudado sobre análise ou design, e tinha participado em projetos desempenhando atividades de analista.

Para diminuir o viés de ter inspetores mais experientes que outros trabalhando a favor de uma das técnicas, os participantes foram distribuídos de forma balanceada. Em outras palavras procurou-se que as equipes de inspeção tivessem a mesma quantidade de inspetores com a mesma experiência. Uma equipe utilizou a técnica *Web DUE* enquanto a outra equipe utilizou a técnica WDP. A Tabela 4-2 apresenta as duas equipes e a experiências dos seus membros. Além dos critérios de classificação mencionados acima, os participantes responderam questões relacionadas a sua

experiência de uso de aplicações *Web*. No entanto, os participantes não foram classificados segundo este critério, pois todos eles possuíam alta experiência.

Houveram outros envolvidos neste estudo experimental, além dos participantes que atuaram como inspetores: (a) o moderador da inspeção, (b) os ministrantes de treinamento das técnicas e (c) a equipe de discriminação. O moderador foi responsável por planejar e coletar os dados durante o estudo experimental. Já os ministrantes dos treinamentos ofereceram uma aula sobre cada uma das técnicas aplicadas. Finalmente, a equipe de discriminação foi responsável por decidir quais dos defeitos apontados eram realmente problemas de usabilidade.

4.2.4 Materiais

Cada equipe avaliou a usabilidade de um conjunto de protótipos em papel de uma aplicação *Web*. No entanto, cada equipe utilizou apenas uma das técnicas, ou a técnica *Web DUE* ou a técnica WDP. Para cada técnica, os participantes receberam uma planilha onde seriam relatados os problemas de usabilidade encontrados. Os *mockups* utilizados neste experimento foram preparados com base em uma aplicação *Web* de compras de cupom. Os sites de compra de cupom oferecem cupons sociáveis que, atualmente, estão surgindo como uma ferramenta de negócios, e como ferramenta para atrair consumidores potenciais [19]. Estes cupons oferecem aos usuários oportunidades de economia na hora de adquirir produtos pela *Web*. Este tipo de aplicação *Web* foi escolhido para servir como objeto a ser avaliado devido ao crescente aumento na sua popularidade [19].

Para auxiliar os inspetores no processo de navegação ao longo dos *mockups* da aplicação *Web*, foi disponibilizado um mapa de navegação. Este mapa indicava que *mockup* devia ser mostrado após a interação com os elementos do *mockup* que o inspetor estava avaliando no momento. A Figura 4-1 apresenta alguns dos *mockups* que foram disponibilizados para os inspetores (ver elementos 1 e 2), assim como um dos mapeamentos utilizados pelos participantes para simular a interação com o sistema (ver elemento 3). Além disso, os *mockups* foram impressos para que os inspetores tivessem uma noção do tamanho real da aplicação na interface. Outro propósito de ter os *mockups* impressos foi que os inspetores podiam apontar diretamente os defeitos neles e fazer anotações, como sugerido pela técnica *Web DUE*.

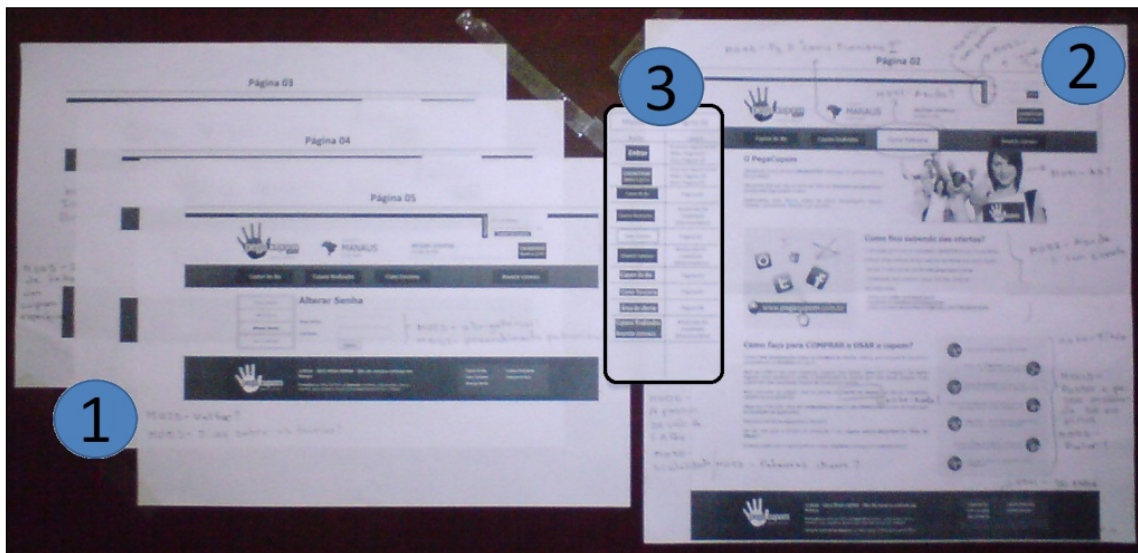


Figura 4-1 Imagens dos *mockups* usados no processo de inspeção e um dos mapeamentos para simular a interação.

4.2.5 Procedimento e Coleta de Dados

A Figura 4-2 apresenta o processo de inspeção utilizado no estudo de viabilidade. Todos os participantes participaram de uma palestra sobre: (a) usabilidade, (b) exemplos de problemas típicos de usabilidade, e (c) como aplicar um método de inspeção como a Avaliação Heurística [28]. Antes de executar a inspeção, cada equipe foi treinada na técnica designada. O contexto da inspeção incluía quatro tarefas que o inspetor devia realizar usando os *mockups* enquanto executada a inspeção com a técnica designada. Cada participante teve um total de três dias para completar a inspeção e preparar um relatório com defeitos de usabilidade. Estes relatórios continham discrepâncias, que são problemas reportados por um inspetor que podem ser um problema real de usabilidade ou um falso positivo. Após finalizar a detecção de problemas, cada inspetor entregava um relatório e um questionário com comentários relacionados à técnica *Web DUE* ou à técnica *WDP*.

Todos os relatórios foram entregues no tempo combinado e nenhum foi descartado. Um dos pesquisadores relacionados com esta pesquisa agiu como moderador. O moderador checkou todas as discrepâncias apontadas nos relatórios dos participantes e as descreveu em um único relatório (ver Figura 4-2). Durante o processo de coleção, todas as discrepâncias duplicadas foram agrupadas e nesses casos manteve-se no relatório a discrepância com a melhor descrição do problema apontado. Este processo foi acompanhado por outro pesquisador para evitar viés.

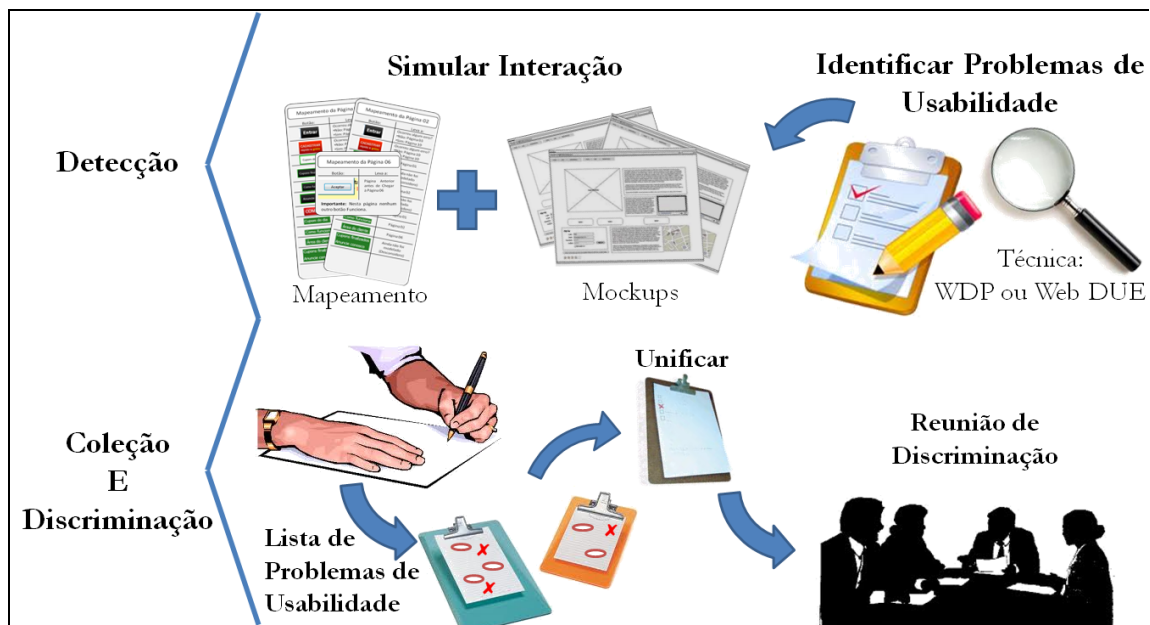


Figura 4-2 Processo de inspeção no estudo de viabilidade.

Após a atividade de coleção, foi feita uma reunião de discriminação. O objetivo da reunião era classificar as discrepâncias do relatório final em defeitos ou falsos positivos. Esta reunião foi feita pelo moderador e outros dois pesquisadores com alta experiência em usabilidade que não estavam envolvidos no estudo ou no desenvolvimento da técnica. Para cada discrepância relatada, os pesquisadores verificavam se este era um problema de usabilidade com base nos *mockups* disponibilizados. Neste processo, o moderador só dirigiu a reunião e não participou do processo de classificação de discrepâncias para evitar viés nas opiniões dos especialistas em usabilidade. Além disso, os pesquisadores que participaram do processo de discriminação não tinham conhecimento de qual técnica tinha sido usada para apontar os defeitos.

Considerando todas as discrepâncias houve um total de 79 defeitos reais identificados. Este número foi usado no cálculo do indicador de eficácia. Na próxima seção serão apresentados os resultados deste estudo de viabilidade e a análise dos mesmos.

4.3 Análise dos Resultados do Estudo de Viabilidade

Neste estudo foram analisados os dados quantitativos e qualitativos obtidos. Os dados quantitativos foram obtidos a partir da lista de discrepâncias resultante da reunião de discriminação. Os resultados qualitativos, por sua vez, foram obtidos a partir das

respostas dos participantes ao questionário. Nas seguintes subseções são apresentadas as análises quantitativas e qualitativas e suas implicações.

4.3.1 Análise Quantitativa

Após a reunião de discriminação, o número de discrepâncias, falsos positivos e defeitos reais foi contabilizado. Além disso, foi considerado o tempo que cada inspetor levou para executar o processo de inspeção dos *mockups*. A Tabela 4-2 apresenta os resultados dos indicadores de eficácia e eficiência por inspetor, e os resultados gerais por técnica. Como mencionado na seção anterior, o número total de defeitos de usabilidade conhecido é de 79.

Tabela 4-2 Resultados do estudo experimental por participante e técnica.

Equipe		Web DUE					WDP		
Participante	1	2	3	4	5	6	7	8	
Resultados da Classificação	Experiência em Usabilidade	M	B	B	A	B	B	A	M
	Experiência em Inspeções	B	B	B	B	B	B	B	A
	Experiência em Análise e Design	B	M	B	A	B	M	M	B
Resultados da Inspeção por Inspetor	Tempo Gasto (min.)	100	67	200	160	147	132	124	279
	Total de Discrepâncias	38	23	27	22	12	33	9	30
	Falsos Positivos	19	2	5	4	1	5	0	8
	Total de Defeitos	19	21	22	18	11	28	9	22
Resultados dos Indicadores por Inspetor	Eficácia (%)	24,05	26,58	27,85	22,78	13,92	35,44	11,39	27,85
	Eficiência (Defeitos por Hora)	11,40	18,81	6,60	6,75	4,49	12,73	4,35	4,73
Resultados dos Indicadores por Técnica	Eficácia Média (%)	25,32					22,15		
	Eficiência Média (Defeitos por Hora)	10,89					6,58		

A análise estatística foi executada usando a ferramenta SPSS v19.0.0 [36]. Nesta análise, considerou-se o valor de $\alpha = 0.10$ devido ao pequeno tamanho da amostra do estudo [9]. Além disso, o método estatístico não paramétrico Mann-Whitney foi utilizado para comparar a eficácia e eficiência das duas amostras. Os gráficos de boxplots das Figuras 4-3 e 4-4 são apresentados para facilitar a visualização dos resultados.

O *boxplot* comparando o indicador de eficácia é mostrado na Figura 4-3. Ao analisar o gráfico, é possível observar que a média do grupo que utilizou a técnica *Web*

DUE foi um pouco maior que a média do grupo que utilizou a técnica WDP. No entanto, a comparação feita usando o teste estatístico Mann-Whitney apontou que não existe diferença significativa entre os grupos ($p = 0.885$). Estes resultados sugerem que a técnica *Web DUE* e a técnica WDP oferecem similares níveis de eficácia quando usadas para executar uma inspeção nos *mockups* de um site de cupons. Estes resultados afirmam a hipótese nula H_{01} e rejeitam a hipótese H_{A1} .

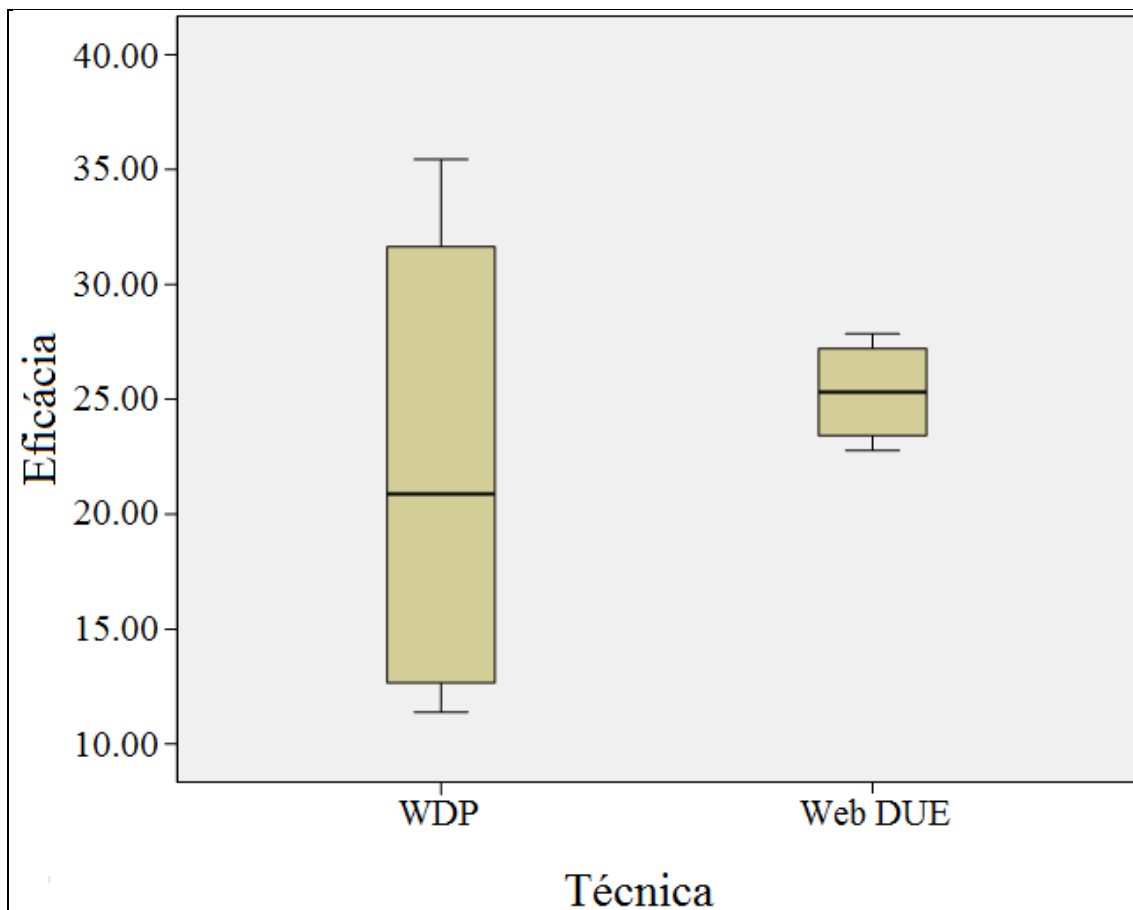


Figura 4-3 Boxplot com o comparativo de eficácia.

O *boxplot* comparando o indicador de eficiência é mostrado na Figura 4-4. Ao analisar o gráfico, é possível observar que a média do grupo que utilizou a técnica *Web DUE* foi maior que a média do grupo que utilizou a técnica WDP. Como consequência, o grupo que fez uso da técnica WDP foi capaz de encontrar mais defeitos em menos tempo. No entanto, a comparação feita usando o teste estatístico Mann-Whitney apontou que não existe diferença significativa entre os grupos ($p = 0.149$). Estes resultados sugerem que: (a) a técnica *Web DUE* e a técnica WDP oferecem diferentes níveis de eficiência quando usadas para executar uma inspeção nos *mockups* de um site

de cupons; e (b) esta diferença não é significativa de um ponto de vista estatístico. Estes resultados afirmam a hipótese nula H_{02} e rejeitam a hipótese H_{A2} .

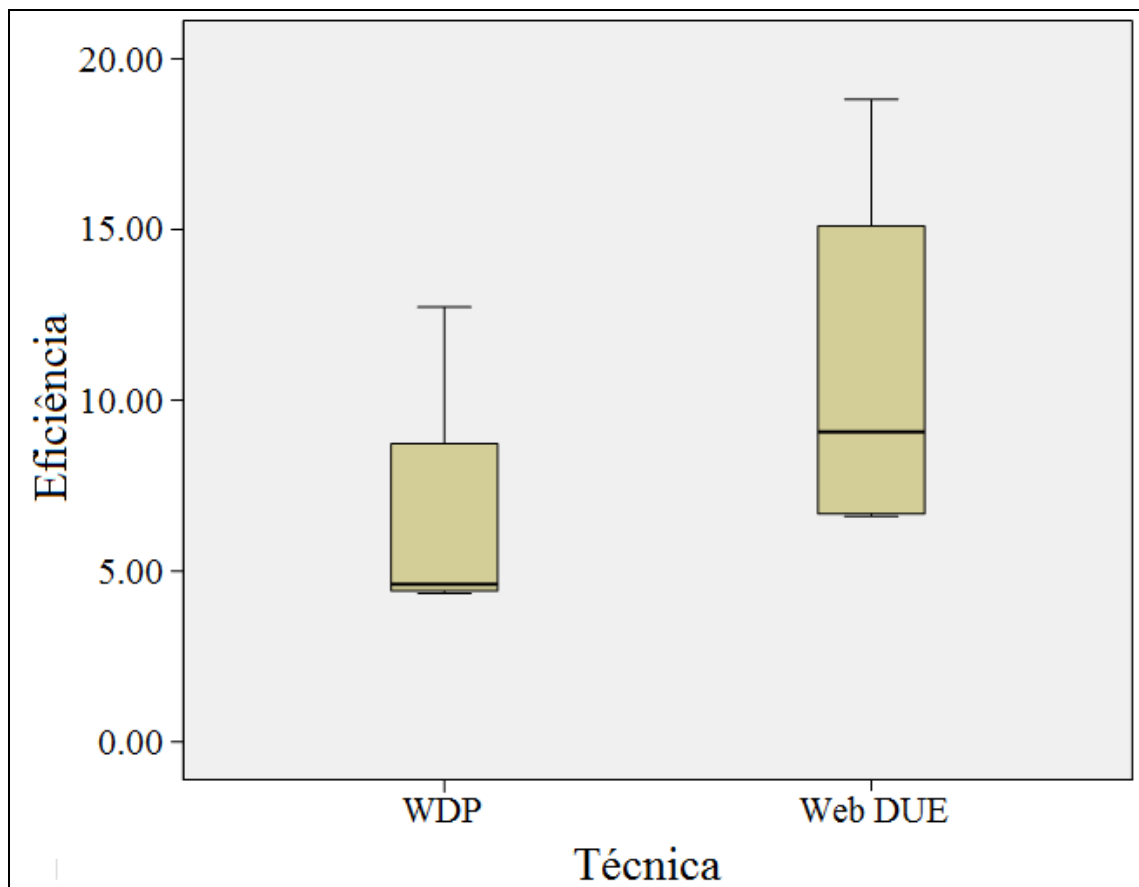


Figura 4-4 Boxplot com o comparativo de eficiência.

Os resultados da análise quantitativa oferecem indícios da superioridade da técnica Web DUE em relação à técnica WDP no que diz respeito à eficiência e eficácia. No entanto, devido ao tamanho da amostra do experimento é necessário realizar mais estudos que comprovem estas observações.

4.3.2 Análise Qualitativa

A análise qualitativa começou com a avaliação das respostas ao questionário disponibilizado aos participantes que utilizaram a técnica *Web DUE*. Os participantes que utilizaram a técnica WDP também responderam a um questionário. No entanto, estas perguntas estavam relacionadas ao uso da técnica WDP para inspecionar *mockups*. Devido ao objetivo do estudo ser a avaliação da viabilidade da técnica *Web DUE* e não a técnica WDP, os resultados relacionados com a técnica WDP não serão discutidos.

A Figura 4-5 apresenta as perguntas do questionário de pós-avaliação da técnica *Web DUE*. As questões tinham por objetivo obter a opinião dos participantes sobre os principais componentes da técnica *Web DUE*: (a) zonas de páginas *Web*, (b) itens de verificação, e (c) exemplos/esclarecimentos dos itens de verificação. Além disso, também foram coletadas informações sobre a adequação e facilidade de uso percebido em relação à técnica. Nesta subseção serão discutidos os resultados qualitativos relacionados à opinião dos participantes em relação ao uso da técnica *Web DUE* para inspeção de *mockups* de aplicações *Web*.

1. Do ponto de vista de inspeção de usabilidade, qual sua opinião em relação à técnica *Web DUE* sobre facilidade de uso? Comente.
2. Do ponto de vista de inspeção de usabilidade, qual sua opinião em relação à adequação da técnica *Web DUE* para inspeção de *mockups*? Comente.
3. As zonas de páginas web propostas (navegação, estado do sistema, ajuda, entrada de dados, etc.) foram descritas de forma que seus significados pudessem ser facilmente entendidos por você? Em caso negativo, o que poderia ser melhorado nas descrições?
4. As zonas de páginas web utilizadas em conjunto com os itens de verificação provocaram algum tipo de dificuldade na identificação de discrepâncias? Comente.
5. Dos exemplos/esclarecimentos disponibilizados para cada item de verificação, existe algum que você não tenha entendido? Em caso afirmativo, quais foram e que poderia ser mudado para melhorar o seu entendimento?
6. Dos exemplos/esclarecimentos disponibilizados para cada item de verificação, existe algum com o que você não tenha concordado? Em caso afirmativo, quais foram e o que você discordou?
7. Se você tivesse que realizar uma nova avaliação de usabilidade de um sistema, seria possível considerar a utilização dessas zonas de páginas web novamente? O que você sugeriria nesta reaplicação da técnica?
8. Em relação ao mapeamento das páginas, comente sobre as facilidades/dificuldades em entender e simular a interação entre o usuário e o sistema. Tem alguma sugestão para simplificar o processo de inspeção de *mockups*?
9. Use o espaço a seguir para comentários gerais que julgar necessários sobre a técnica, sua aplicação, dificuldades encontradas, etc.

Figura 4-5 Questionário pós-avaliação da técnica *Web DUE*.

Com relação à opinião dos participantes sobre o uso de zonas de páginas *Web*, todos os inspetores concordaram que foi fácil entender e identificar as zonas contidas nos *mockups*. Este fato é ilustrado pelas seguintes citações:

“Sim, foram muito esclarecedoras; não tive problemas em entendê-las.”

- Inspetor 1.

“Foi muito fácil entender todas as zonas de páginas Web.”

- Inspetor 2.

O inspetor 4 declarou que talvez seria mais fácil executar a inspeção se a técnica *Web DUE* permitisse a verificação de zonas de páginas *Web* relacionadas dentro de uma página *Web*. Em outras palavras, se um inspetor olhasse a mesma zona de páginas *Web* mais de uma vez dentro do *mockup*, ele deve ser capaz de verificar todos os itens para cada zona repetida. Avaliar o mesmo *checklist* várias vezes pode ter causado um incremento no tempo gasto durante a inspeção.

“No caso de uma zona de entrada de dados, uma página Web pode conter mais de uma dessas zonas para diferentes propósitos. A técnica podia permitir a avaliação desses itens de verificação repetidos de uma forma especial.”

- Inspetor 4.

Com relação aos itens de verificação da técnica *Web DUE*, os inspetores 1, 3 e 4 concordaram com que os itens ajudavam quando combinados com as zonas de páginas *Web* (ver citação do inspetor 4 a seguir). No entanto, o inspetor 2 apontou que alguns itens de verificação podem ser usados para avaliar a mesma zona de páginas *Web* (ver citação do inspetor 2 a seguir). Isto significa que alguns itens podem ser usados para avaliar os mesmos componentes, o que pode confundir os inspetores, ou que alguns dos itens de verificação são muito abrangentes para permitir a identificação de problemas específicos da zona de páginas *Web* inspecionada. Além disso, o inspetor 3 afirmou que alguns itens de verificação não ajudavam a decidir se havia ou não um problema de usabilidade (ver citação do inspetor 3 a seguir). Isto significa que alguns itens de verificação podem não estar provendo descrições claras de quais não conformidades devem ser identificadas, ou que os itens de verificação são muito subjetivos para que um inspetor consiga decidir.

“..., os itens de verificação contribuíram para encontrar mais discrepâncias ao analisar uma determinada parte da página.”

- Inspetor 4.

“A zona de entrada de dados, por exemplo, possui itens de verificação que podem ser adequados para a avaliação da zona de ajuda. Esta ambiguidade faz com que seja difícil encontrar termos que definam os problemas encontrados durante a inspeção.”

- Inspetor 2.

“... No entanto, alguns itens de verificação julgam atributos que podem não estar totalmente certos ou errados.”

- Inspetor 3.

Com relação à opinião dos inspetores com respeito aos exemplos e esclarecimentos disponibilizados junto com os itens de verificação, todos os inspetores concordaram que eram fáceis de entender. Além disso, eles afirmaram que os exemplos e esclarecimentos também facilitavam o entendimento dos itens de verificação.

Quando os inspetores foram questionados em relação à adequação da técnica, um inspetor (Inspetor 3) discordou da sua adequação para a inspeção de *mockups*. No entanto, este inspetor não estava se referindo à técnica e sim ao processo de mapeamento dos *mockups*. Todos os inspetores concordaram em que a técnica podia apoiar a identificação de problemas de usabilidade em *mockups*. No entanto, devido o fato dos inspetores ter de simular manualmente a interação entre o usuário e o sistema tornou o processo de inspeção muito cansativo. As seguintes citações ilustram a opinião dos inspetores respeito à dificuldade de simular a interação:

“... Aplicar a técnica em mockups é muito confuso quando se navega pelas páginas. Particularmente em sistemas maiores, com um número maior de páginas. Além disso, não existe interação com o sistema.”

- Inspetor 3.

“... Foi muito difícil simular usando páginas Web. Navegar entre as páginas é muito complexo, mesmo com o uso do mapa de navegação.”

- Inspetor 3.

Com relação à facilidade de uso percebida, o inspetor 4 achou a técnica muito difícil. O inspetor apontou que a técnica era muito detalhada e repetitiva (ver citação do inspetor 4). Os outros inspetores não apontaram que a técnica era difícil, mas que o processo de simular a interação era difícil, como descrito anteriormente.

“... A técnica não me pareceu nada fácil. Era muito detalhada e repetitiva durante a maior parte da avaliação. Por outro lado, inclui uma visão completa dos conteúdos das páginas.”

- Inspetor 4.

Os inspetores também citaram vantagens e sugestões com respeito à avaliação de protótipos em papel. Segundo o inspetor 4, o uso de *mockups* permite aos inspetores apontarem os problemas de usabilidade diretamente (ver citação do inspetor 4). Com

relação a sugestões para a inspeção de *mockups*, o inspetor 3 indicou que poderiam ser usados *mockups* coloridos para deixar os *mockups* mais realísticos (ver citação do inspetor 3). Além disso, o inspetor 2 disse que a organização dos *mockups* é importante para simular a interação entre o usuário e o sistema (ver citação do inspetor 2).

“Eu achei a técnica apropriada pois nos mockups podemos riscar e depois relatar o que desenhamos nas opções dos checklists.”

- Inspetor 4.

“Para uma visão mais artística/visual do site, poderiam se usar mockups coloridos.”

- Inspetor 3.

“O uso de um grafo explicando a interação entre as páginas, ou a organização das páginas de acordo com as atividades que estão sendo executadas poderia ajudar a entender o processo de interação.”

- Inspetor 2.

No geral, os resultados mostram que a técnica *Web DUE* pode ser aplicada para avaliar *mockups* de aplicações *Web*. Mesmo assim, os resultados qualitativos indicam que ainda existem possibilidades de melhoria. Na seguinte seção, são apresentadas as ameaças à validade do estudo de viabilidade da técnica *Web DUE*.

4.4 Análise dos Resultados do Estudo de Viabilidade

Todo estudo experimental possui ameaças que podem afetar a validade dos seus resultados. Nas seguintes subseções são apresentadas as ameaças à validade consideradas neste estudo. Estas ameaças foram agrupadas em quatro principais categorias: interna, externa, conclusão e constructo [40].

4.4.1 Validade Interna

Segundo Wohlin *et al.* [40], a validade interna avalia se, de fato, o tratamento causa os resultados. Neste estudo foram consideradas quatro principais ameaças à validade interna: (a) efeitos de treinamento, (b) conhecimento em usabilidade dos participantes, (c) classificação dos inspetores, e (d) medição do tempo. Com relação aos efeitos do treinamento, pode haver uma ameaça à validade se a qualidade do treinamento da técnica WDP tivesse sido inferior à qualidade do treinamento da técnica *Web DUE*. No entanto, esta ameaça foi controlada ao utilizar os mesmos exemplos de problemas de

usabilidade em ambos os treinamentos. Além disso, para mitigar a ameaça do conhecimento dos participantes, estes foram divididos em equipes balanceadas segundo sua experiência. Esta medida evitou que a experiência dos participantes afetara os resultados obtidos por técnica. Outro problema pode ter sido a classificação dos participantes. Na Tabela 4-2 pode ser observado que os participantes foram classificados de acordo com três critérios objetivos. Para diminuir o viés de classificação, o grau de experiência em cada critério foi definido com base nos dados de um formulário de caracterização objetivo. Finalmente, com relação à medição do tempo, foi solicitado aos participantes que fossem precisos nas suas medições. No entanto, não pode ser garantido que estas medidas foram obtidas de forma cuidadosa.

4.4.2 Validade Externa

A validade externa se preocupa com a generalização dos resultados [40]. Nesse sentido, serão discutidos quatro problemas relacionados à validade externa deste estudo: (a) estudantes não são provavelmente bons substitutos de inspetores profissionais, (b) ambientes acadêmicos não representam a experiência do dia a dia da indústria, (c) se um site de cupons é uma aplicação representativa de todos os tipos de aplicações *Web*, e (d) se a técnica WDP é uma técnica adequada para comparação. Com respeito ao primeiro problema, o uso de estudantes como inspetores, pode se dizer que como alguns dos estudantes tinham experiência na indústria, estudantes podiam sim substituir inspetores experientes. Além disso, nenhum deles tinha tido contato com as técnicas utilizadas. Apesar do estudo ter sido executado em um ambiente acadêmico, os *mockups* utilizados foram baseados em uma aplicação *Web* real, o que pode ajudar a simular um ambiente real de desenvolvimento de software. Com relação ao uso de um site de cupons, não pode se garantir que esta não seja uma ameaça, pois existem vários tipos de categorias de aplicações *Web* [18]. Finalmente, o uso da técnica WDP com fins comparativos é uma ameaça à validade. No entanto, pode ser afirmado que, uma vez que a técnica *Web DUE* está baseada na técnica WDP, é razoável comparar se a técnica *Web DUE* apresenta melhores resultados que a técnica WDP ao avaliar *mockups*.

4.4.3 Validade de Conclusão

Segundo Wohlin *et al.* [40], a validade de conclusão se preocupa com o relacionamento entre o tratamento e os resultados. Neste estudo, o maior problema é o poder estatístico. Uma vez que o número de participantes é baixo, os dados extraídos neste estudo só

podem ser considerados indicadores da viabilidade da técnica *Web DUE* e não conclusivos.

4.4.4 Validade de Constructo

A validade de constructo analisa o relacionamento entre a teoria e a observação [40]. Os critérios usados para medir a viabilidade da técnica podem ser considerados uma ameaça se não foram corretamente escolhidos. Como eficácia e eficiência são dois critérios comuns para a investigação da produtividade de novas técnicas, esta ameaça não pode ser considerada um risco para a validade dos resultados [11]. Além disso, estes critérios foram medidos usando a mesma abordagem proposta em [6].

4.5 Considerações Finais

Este capítulo discutiu os resultados de um estudo experimental para responder a seguinte questão de pesquisa: “A técnica *Web DUE* é viável em termos do número de defeitos encontrados e é o tempo bem gasto?” Para responder a esta questão de pesquisa foram analisados os resultados da técnica *Web DUE* em termos de eficácia e eficiência, e comparados com os resultados da técnica base, a técnica WDP. Ambos os resultados quantitativos e qualitativos deste estudo ofereceram *feedback* importante para melhorar a técnica *Web DUE*.

A análise dos dados quantitativos usando o método estatístico Mann-Whitney apontaram que o indicador de eficácia é similar em ambas as técnicas. No entanto, a técnica *Web DUE* conseguiu melhores resultados com 25,32% em comparação com os 22,15% da técnica WDP. Além disso, a análise do indicador de eficiência, usando o mesmo método estatístico, apontou que também não existe diferencia significativa no indicador de eficácia. No entanto, a técnica *Web DUE* atingiu uma taxa de 10,89 defeitos por hora enquanto a técnica WDP um total de 6,58 defeitos por hora. Estes resultados devem ser considerados indicativos, mas não conclusivos, da viabilidade da técnica *Web DUE*.

A análise qualitativa mostrou que, no geral, os inspetores acharam a técnica *Web DUE* fácil de aplicar e adequada para a avaliação de *mockups* de aplicações *Web*. Também indicaram a habilidade de guiar o processo de inspeção por zonas próprias de páginas *Web* podia ajudar no desempenho da avaliação. Além disso, foi apontado que os

itens de verificação, combinados com os exemplos e esclarecimentos da técnica *Web DUE* podiam contribuir a encontrar mais problemas de usabilidade.

Apesar do *feedback* positivo, os inspetores deram sugestões de melhoria no processo de inspeção de *mockups* de aplicações *Web*. Os inspetores apontaram que o maior problema com este tipo de avaliação é o cansaço provocado pelo processo de simulação da interação entre o usuário e o sistema. Como consequência desta afirmativa, foi proposto o apoio ferramental da técnica *Web DUE*, a ferramenta *Mockup Design Usability Evaluation (Mockup DUE)*. Esta ferramenta pretende abstrair o processo de simulação da interação, permitindo ao inspetor focar diretamente na avaliação de usabilidade.

Os inspetores afirmaram que o uso de *mockups* coloridos podia ajudar a disponibilizar uma visão mais artística dos *mockups*. Esta sugestão foi levada em consideração e adicionada à lista de requisitos do apoio ferramental. Além disso, os inspetores apontaram que as principais vantagens do uso de *mockups* como artefatos a serem inspecionados é que eles permitem ao inspetor apontar o problema diretamente e fazer anotações. Esta característica será mantida na proposta do apoio ferramental da técnica *Web DUE*.

No seguinte capítulo, é apresentada a proposta do apoio ferramental da técnica *Web DUE*. A ferramenta *Mockup DUE*, como descrito acima visa diminuir a carga sobre o inspetor ao simular a interação entre o usuário e o sistema enquanto o inspetor executa a avaliação de usabilidade em cima dos *mockups*. A ferramenta foi avaliada através de um estudo qualitativo que será descrito no Capítulo 6.

5 A Ferramenta *Mockup Design Usability Evaluation*

Com o objetivo de reduzir o esforço dos inspetores ao aplicarem a técnica Web DUE, foi proposta a ferramenta Mockup Design Usability Evaluation (Mockup DUE). Neste capítulo são apresentadas as funcionalidades da ferramenta com base nos resultados do estudo de viabilidade apresentado no capítulo anterior.

5.1 Introdução

Com o intuito de diminuir os custos de correção dos problemas de usabilidade, foi proposta a técnica *Web Design Usability Evaluation (Web DUE)*. Esta técnica permite a inspeção de *mockups* e guia os inspetores pelo processo de inspeção por meio de zonas próprias de páginas *Web* [13]. No entanto, no estudo de viabilidade relatado no capítulo anterior, descobriu-se que os inspetores tinham dificuldade em utilizar a técnica, pois a mesma requer que o inspetor realize o mapeamento manual dos *mockups*. Isto é, o próprio inspetor deve simular a interação com o sistema enquanto executa a inspeção.

Para diminuir a sobrecarga do inspetor durante o processo de identificação de problemas, surgiu a ideia de criar uma ferramenta de apoio ao processo de inspeção da técnica *Web DUE*. Esta ferramenta foi chamada de *Mockup Design Usability Evaluation (Mockup DUE)* e abstrai o processo de mapeamento dos *mockups* para que o inspetor possa se concentrar na identificação de problemas.

Neste capítulo são apresentadas, em detalhe, as características e funcionalidades suportadas pela ferramenta *Mockup DUE*. Na Seção 5.2 são relacionados os resultados do estudo de viabilidade do capítulo anterior com as funcionalidades a serem suportadas pela ferramenta. Na Seção 5.3, é apresentada a primeira versão da ferramenta e as etapas suportadas relacionadas com o processo de inspeção da técnica *Web DUE*. Finalmente a Seção 5.4 apresenta as considerações finais deste capítulo.

5.2 A Concepção da Ferramenta *Mockup DUE*

Os resultados qualitativos do primeiro estudo de viabilidade da técnica *Web DUE* indicaram que os inspetores consideraram a técnica útil e apropriada para a inspeção de *mockups* de aplicações *Web*. Além disso, os inspetores afirmaram que os itens de verificação de usabilidade da técnica, combinados com as zonas próprias de páginas *Web*, podiam simplificar e apoiar a identificação de problemas de usabilidade. No entanto, o fato dos inspetores terem de simular manualmente a interação entre o usuário e o sistema tornou a técnica muito cansativa. Além disso, os inspetores apontaram que o uso de *mockups* como artefatos *Web* a serem inspecionados permitia: (a) apontar o defeito diretamente no *mockup*, e (b) adicionar notas. Finalmente, os inspetores sugeriram utilizar *mockups* coloridos para que seja possível visualizar mais detalhes da aplicação que não pudessem ser percebidos em *mockups* preto e branco.

No Capítulo 3, foi mencionado que a automação é uma das características relevantes dos UIMs de aplicações *Web* emergentes (ver Subseção 3.2). Esta característica se refere a proporcionar meios pelos quais o esforço de realizar uma inspeção seja diminuído. Nem todas as técnicas de inspeção podem ser totalmente automatizadas, sobretudo quando dependem da experiência e opinião do inspetor. No caso da técnica *Web DUE* deve ser providenciado um apoio ferramental que facilite o processo de inspeção para melhorar o desempenho dos inspetores.

Levando em consideração as informações obtidas no estudo de viabilidade e com o propósito de apoiar a indústria de desenvolvimento de software no que diz respeito a UIMs para aplicações *Web*, surgiu a ideia de implementar a ferramenta *Mockup Design Usability Evaluation (Mockup DUE)*. A ferramenta *Mockup DUE* foi desenvolvida para facilitar o processo de inspeção da técnica *Web DUE* sem perder as características que faziam da inspeção de *mockups* uma proposta atraente. Nesse sentido a ferramenta *Mockup DUE* possui as seguintes funcionalidades:

- Permite o mapeamento de *mockups*, ou seja, relacionar os elementos dos *mockups* com os *mockups* que devem ser apresentados.
- Permite a simulação da interação entre o usuário e o sistema durante o processo de detecção de problemas de usabilidade.
- Permite navegar entre os *mockups* com flexibilidade, sendo possível visualizar qualquer um dos *mockups* quando desejado.
- Permite apontar defeitos de usabilidade diretamente no *mockup*.

- Permite adicionar notas diretamente no *mockup*.
- Permite utilizar a técnica *Web DUE* para encontrar os problemas de usabilidade.
- Permite gerar um relatório contendo os defeitos identificados após o processo de detecção de problemas.
- Permite salvar e abrir posteriormente os *mockups* inspecionados.
- Permite contar o tempo de inspeção para avaliar o desempenho do inspetor ao utilizar a técnica *Web DUE* em conjunto com a técnica.

Na próxima seção serão mostradas imagens da primeira versão da ferramenta *Mockup DUE* assim como os passos a serem seguidos para realizar uma avaliação em conjunto com a Técnica *Web DUE*.

5.3 Aplicando a Primeira Versão da Ferramenta *Mockup DUE*

A ferramenta *Mockup DUE* foi desenvolvida utilizando o framework de desenvolvimento QT². O QT é um framework multiplataforma que permite desenvolver software livre e pode rodar em várias plataformas como Windows, Linux e Macintosh. A vantagem do uso de QT é que possui ferramentas de criação de interfaces de usuário que facilitam o desenvolvimento de interfaces rápidas e interativas [1]. Esta característica é importante, pois se tratando de uma ferramenta de apoio ao processo de inspeção da técnica *Web DUE*, a mesma deve possuir boa usabilidade.

A tela inicial da ferramenta *Mockup DUE* é mostrada na Figura 5-1. Como pode ser observado na figura, a ferramenta atua em duas etapas principalmente: planejamento da inspeção e detecção de problemas de usabilidade. Durante a etapa de planejamento, os moderadores da inspeção, que preparam os *mockups* para inspeção devem carregar os *mockups* e ligá-los por meio de links. Um link para a ferramenta *Mockup DUE* serve para levar o usuário a outro *mockup*. Os inspetores podem utilizar estas áreas clicáveis para simular a interação entre o usuário e o sistema e realizar tarefas usando os *mockups*. Esta característica permite criar *mockups* mais reais com a aplicação final real.

² <http://qt.digia.com/>

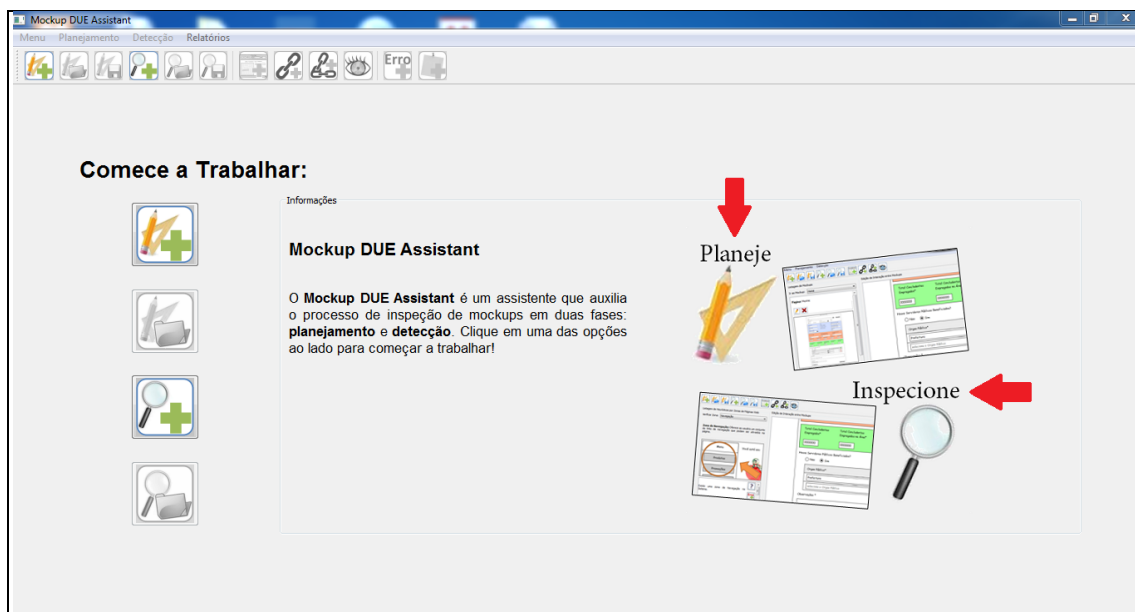


Figura 5-1 Tela inicial da ferramenta *Mockup DUE*.

Durante a etapa de detecção de problemas, os inspetores utilizam os *mockups* que foram mapeados pelos moderadores e realizam uma inspeção usando a técnica *Web DUE*. Nesta etapa, o sistema disponibiliza as zonas de páginas *Web* e os itens de verificação para as zonas. O inspetor deverá então interagir com os *mockups* e apontar os defeitos caso um dos itens de verificação seja violado. Nesta etapa também é possível anotar o tempo da inspeção, fazer anotações nos *mockups* e gerar um relatório contendo todos os problemas identificados.

Nas seguintes subseções será detalhada cada uma das etapas do processo de inspeção da técnica *Web DUE* fazendo uso da ferramenta. Para facilitar o entendimento de cada etapa também serão disponibilizadas imagens das telas da ferramenta durante seu uso.

5.3.1 Planejamento da Inspeção

Para realizar uma inspeção utilizando a técnica *Web DUE* os inspetores devem receber um conjunto de *mockups* mapeados. Este mapeamento de *mockups* acontece durante o processo de planejamento da inspeção. Nesta etapa os moderadores identificam que elementos da interface podem gerar uma transição de um *mockup* para outro e adicionam links na ferramenta *Mockup DUE*. Para iniciar um novo planejamento os inspetores clicam nos botões marcados na Figura 5-2. Assim que o moderador passa o mouse em um dos botões, a ferramenta apresenta informações sobre o processo de

planejamento. Estas informações são importantes para moderadores ou inspetores novatos no uso da ferramenta *Mockup DUE*.

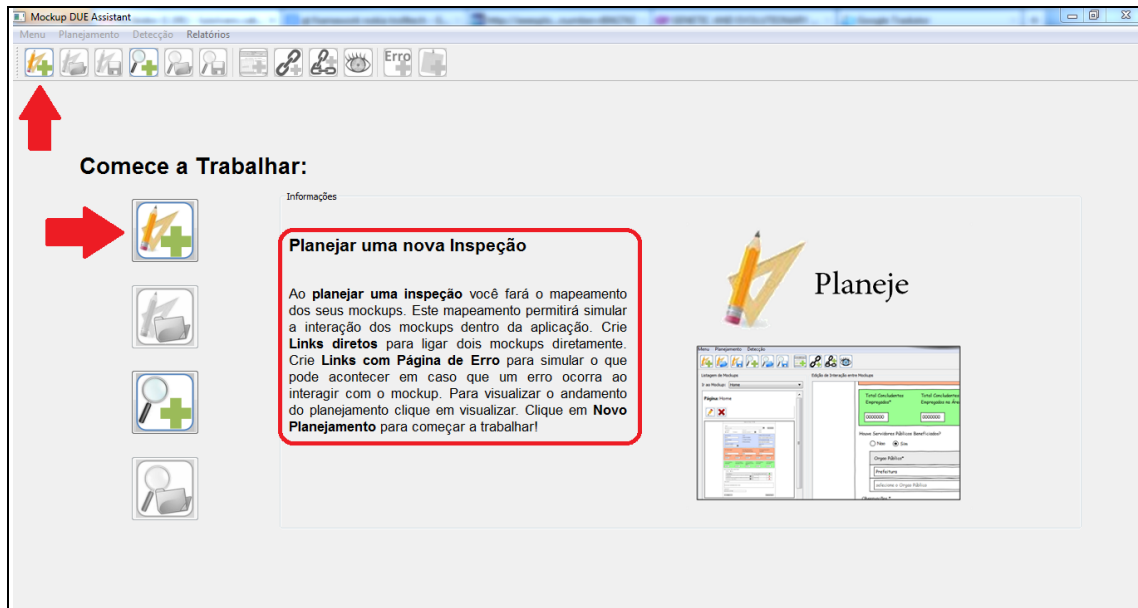


Figura 5-2 Iniciando um planejamento de inspeção.

Ao clicar no botão de novo planejamento, a ferramenta carrega a tela da Figura 5-3. Essa tela é dividida em duas partes: (a) a listagem de *mockups* e (b) a visualização/mapeamento de *mockups*. Na primeira parte, o moderador pode visualizar todos os *mockups* que foram cadastrados, enquanto na segunda, o moderador pode adicionar links nos *mockups* para simular a interação entre eles.

Para ativar as funcionalidades relacionadas com o mapeamento dos *mockups* o moderador pode clicar nos botões da barra de ferramentas (ver elementos numerados na Figura 5-3). O botão nº 1 serve para cadastrar um novo *mockup*. Quando o moderador clicar o botão, o sistema apresentará a tela de cadastro de *mockups* (ver Figura 5-4 parte 1). Nessa tela o usuário terá de informar o nome e descrição do *mockup* assim como carregar uma imagem. Uma vez que o *mockup* for cadastrado o sistema atualizará a listagem e apresentará o *mockup* para adicionar links. O moderador pode então usar os botões nº 2 e nº 3. Estes botões permitem cadastrar links diretos e links com páginas de erro. A tela de cadastro de link direto é mostrada na Figura 5-4 (ver parte B). Um link direto é aquele que liga um *mockup* com outro apenas é clicado. Já um link com página de erro permite ao inspetor escolher entre dois cenários: (a) positivo, onde a interação foi bem sucedida e não aconteceu nenhum erro, e (b) negativo, onde o sistema apresentou um erro. Nesse caso, ao cadastrar o link com página de erro (ver Figura 5-4 parte C), o moderador deverá ter cadastrado também o *mockup* a ser apresentado em

caso de erro do sistema. Este *mockup* pode conter uma mensagem de erro ou alguma resposta do sistema para informar ao usuário que aconteceu algum problema ao processar as informações. Na parte D da Figura 5-4 foram adicionados dois links, um link direto (ver Figura 5-4 elemento 1) e um link com página de erro (ver Figura 5-4 elemento 2). Para que o inspetor possa diferenciar entre os diferentes tipos de link, optou-se pela utilização de cores diferentes. Links diretos são azuis enquanto links com página de erro são vermelhos. Além disso, os links são feitos em marca de água para não atrapalhar a visualização dos elementos do *mockup*.

Para permitir flexibilidade na hora de cadastrar links, estes também podem ser redimensionados. O usuário deve colocar o mouse em cima deles e o sistema apresentará quatro pontos de redimensionamento (ver Figura 5-4 elemento 1). O usuário da ferramenta pode arrastar estes pontos e mudar o tamanho do link. Além disso, ao clicar no link o usuário pode posicioná-lo no local desejado, é por isso que se têm dois links em partes diferentes do *mockup* na Figura 5-4. O moderador também pode modificar o destino do link, ou apagá-lo apertando os botões de “editar” e “deletar link”. Estes botões são mostrados no elemento 1 da Figura 5-4, e só aparecem quando o usuário passa o mouse em cima do link. Além disso, passar o mouse em cima também mostra um tooltip com informações sobre o destino do link (ver Figura 5-4 elemento 3).

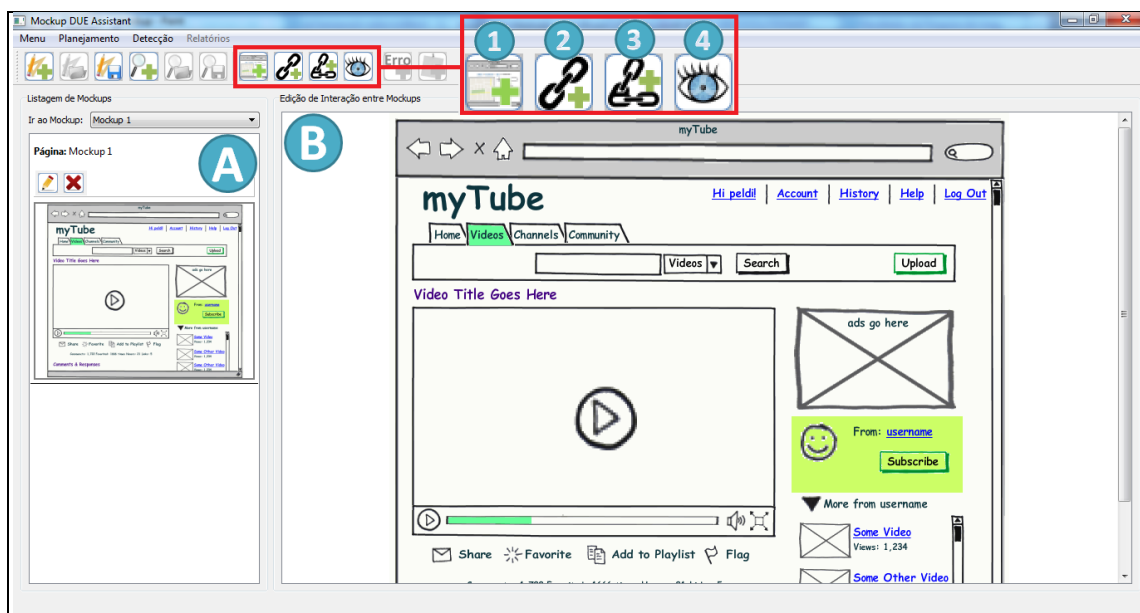


Figura 5-3 Tela de planejamento de inspeção.

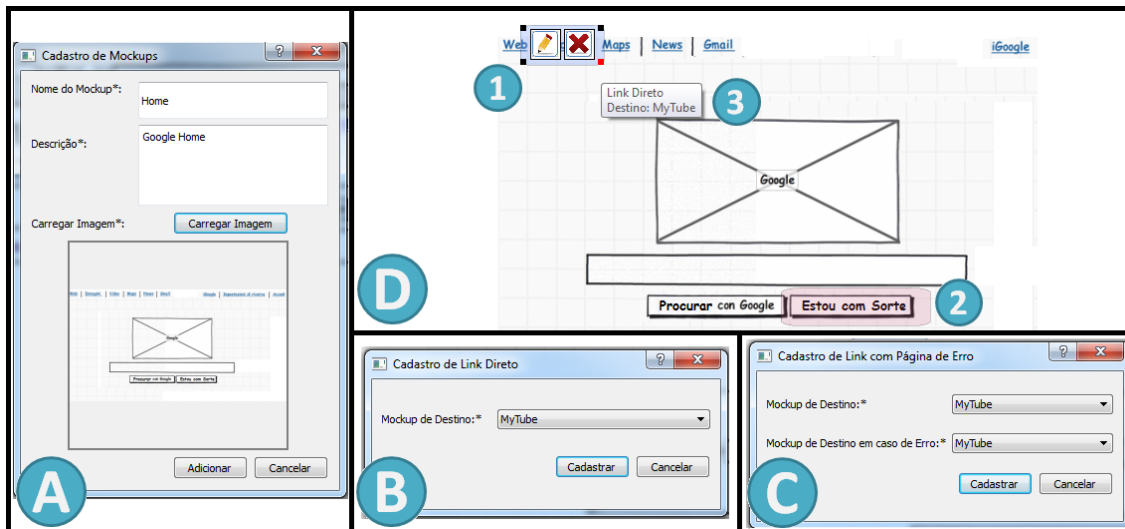


Figura 5-4 Funcionalidades do planejamento de inspeção.

Os inspetores podem verificar se a interação foi modelada de forma correta apertando no botão nº 4 (ver Figura 5-3). Este botão apresenta uma visão prévia dos *mockups* mapeados. Uma vez que o mapeamento dos *mockups* tiver sido finalizado, os moderadores poderão salvar o planejamento em um arquivo XML para que este possa ser usado pelos inspetores durante o processo de detecção de defeitos. Na seguinte subseção, será apresentada a etapa de detecção de defeitos usando a ferramenta *Mockup DUE*.

5.3.2 Detecção de Problemas e Relatório

Como descrito na subseção anterior, uma vez feito o mapeamento dos *mockups*, os inspetores podem começar a avaliar a sua usabilidade usando a técnica *Web DUE* que é apresentada na ferramenta *Mockup DUE*. Nesta etapa, os inspetores carregam os dados de um planejamento de inspeção previamente salvo, e utilizam os itens de verificação da técnica *Web DUE* para identificar problemas de usabilidade. Para iniciar uma nova detecção os inspetores clicam nos botões marcados na Figura 5-5. Assim, como na etapa de planejamento, a ferramenta apresenta informações sobre o processo de detecção quando o inspetor passa o mouse pelos botões.

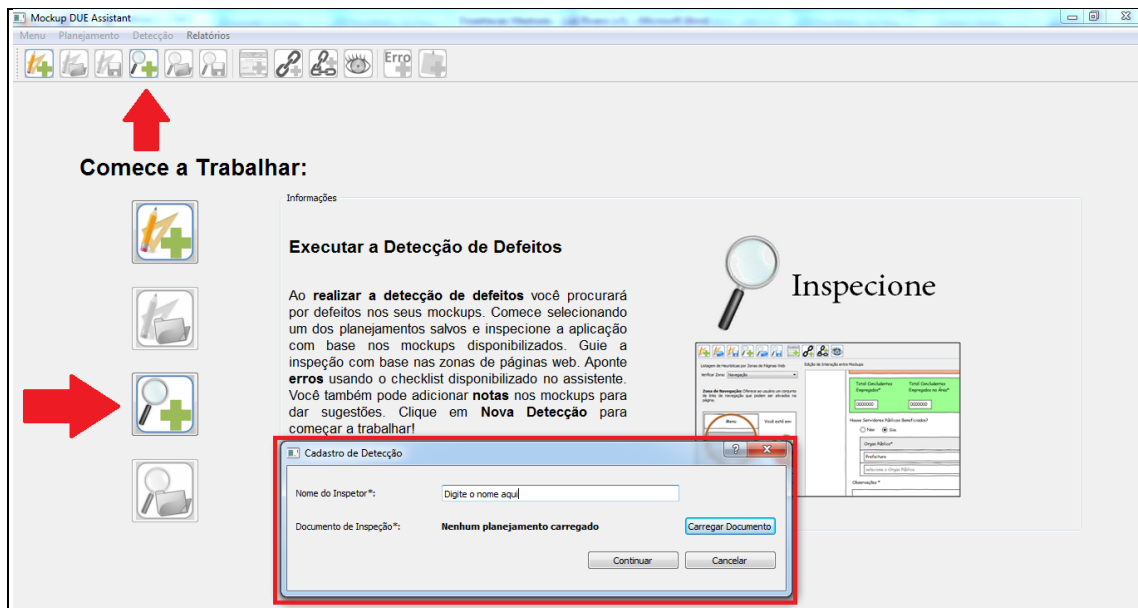


Figura 5-5 Iniciando o processo de detecção de problemas de usabilidade.

Quando o inspetor clicar num dos botões da Figura 5-5, a ferramenta solicitará os dados do inspetor. Estes dados são importantes para manter o controle de quem está realizando a detecção de defeitos. Além disso, o sistema pedirá ao inspetor que carregue um planejamento de inspeção. Este planejamento contém o mapeamento dos *mockups* descrito na subseção anterior.

Uma vez que o inspetor preenche os dados e carrega o planejamento, o sistema apresenta a tela de detecção de defeitos. Esta tela é apresentada na Figura 5-6 e possui duas partes principais: (a) os *checklists* de verificação da técnica *Web DUE* agrupados por zonas próprias de páginas *Web*, e (b) a visualização de *mockups* conforme o mapeamento definido no planejamento da inspeção. Na parte A da Figura 5-6, o inspetor tem acesso às zonas de páginas *Web*, ou seja, pode visualizar a descrição das zonas, um exemplo para verificá-las, e todos os itens de verificação relacionados. Na parte B da Figura 5-6, por sua vez, permite visualizar os *mockups* mapeados e interagir com eles. Além disso, os links que foram adicionados no planejamento da subseção anterior continuam presentes nos *mockups*. Estes links serão utilizados pelos inspetores para navegar pelos *mockups*. Se o inspetor quiser acesso direto a um *mockup*, basta selecionar um dos *mockups* do *combobox* “ir ao Mockup:”. Este *combobox* mostra o *mockup* instantaneamente.

O tempo gasto durante a inspeção é um dos dados relevantes para o cálculo do indicador de eficiência da técnica. Por esse motivo, para posteriores análises quantitativas, foi adicionado um relógio para obter esse dado de forma automática.

Além disso, para ser mais preciso, o inspetor pode pausar o relógio quando tiver parado de realizar a inspeção, ou continuar a contagem do tempo, quando continuar a inspeção.

Para ativar as funcionalidades relacionadas com a detecção de defeitos de usabilidade, o inspetor pode clicar nos botões da barra de ferramentas (ver elementos numerados na Figura 5-6). Ao clicar no botão nº 1 o inspetor pode tirar dúvidas sobre um determinado item de verificação. Na Figura 5-7 parte A, a ferramenta *Mockup DUE* apresenta os exemplos/esclarecimentos relacionados com o item de verificação “existe uma zona de estado do sistema”. Ao clicar nos botões nº 2 e nº 3, o inspetor estará cadastrando um defeito. No entanto, este cadastro pode ser feito de formas diferentes: diretamente, ou através dos itens de verificação. Quando o inspetor clica no botão nº 2 se pressupõe que o inspetor é mais experiente no uso da ferramenta e da técnica *Web DUE*. Consequentemente, o inspetor não precisa verificar todos os itens de verificação e sim apontar os defeitos que já identificou. Para isso, a ferramenta *Mockup DUE* tem um cadastro especializado (ver Figura 5-7 parte B) que disponibiliza todas as zonas, e para a zona selecionada os itens de verificação relacionados. O inspetor pode selecionar as zonas e os itens de verificação violados, escrever suas considerações e cadastrar o defeito. A outra forma de cadastrar um defeito é clicando diretamente no botão “cadastrar defeito” do lado do item de verificação violado (ver Figura 5-6 botão nº 3). Esse cadastro alternativo é feito para inspetores novatos com pouca experiência no uso da técnica *Web DUE* e da ferramenta *Mockup DUE*. Nesse cadastro, a ferramenta automaticamente preenche os campos para o inspetor e este só precisa clicar em cadastrar e adicionar algum comentário se achar necessário (ver Figura 5-7 parte C). Finalmente, usando o botão nº 4 da Figura 5-6, o inspetor pode adicionar notas para dar sugestões, ou esclarecer alguma coisa que considere necessário. Quando este botão é acionado, o sistema apresenta a tela de cadastro de notas (ver Figura 5-7 parte D). O usuário escreve suas anotações e salva a nota no *mockup* correspondente.

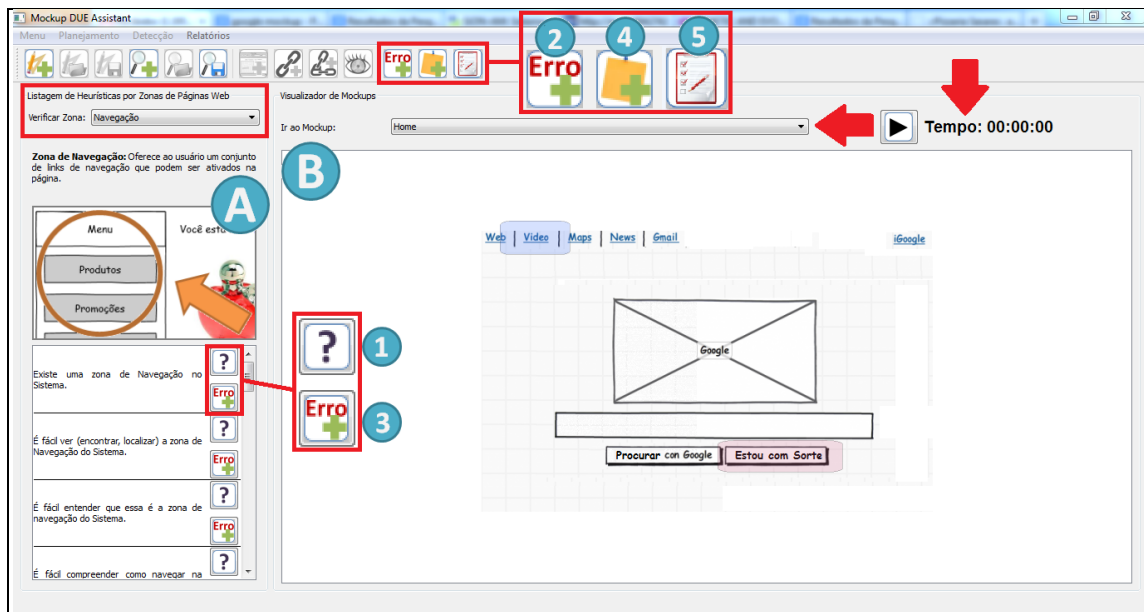


Figura 5-6 Tela de detecção de problemas de usabilidade.

A Figura 5-7 parte E mostra como foram cadastrados dois defeitos e uma nota. Vale ressaltar que estes são somente exemplos de cadastro e não de problemas de usabilidade ou anotações reais. Sendo assim, um defeito é representado como uma elipse e pode ser redimensionado e reposicionado seguindo o mesmo procedimento no cadastro de links (ver Figura 5-7 elemento 1). Já as notas não são redimensionadas, porém só reposicionadas no local desejado (ver Figura 5-7 elemento 3). O sistema também permite ao usuário visualizar as informações cadastradas, sejam defeitos ou anotações, passando o mouse por cima (ver Figura 5-7 elementos 2 e 4 respectivamente).

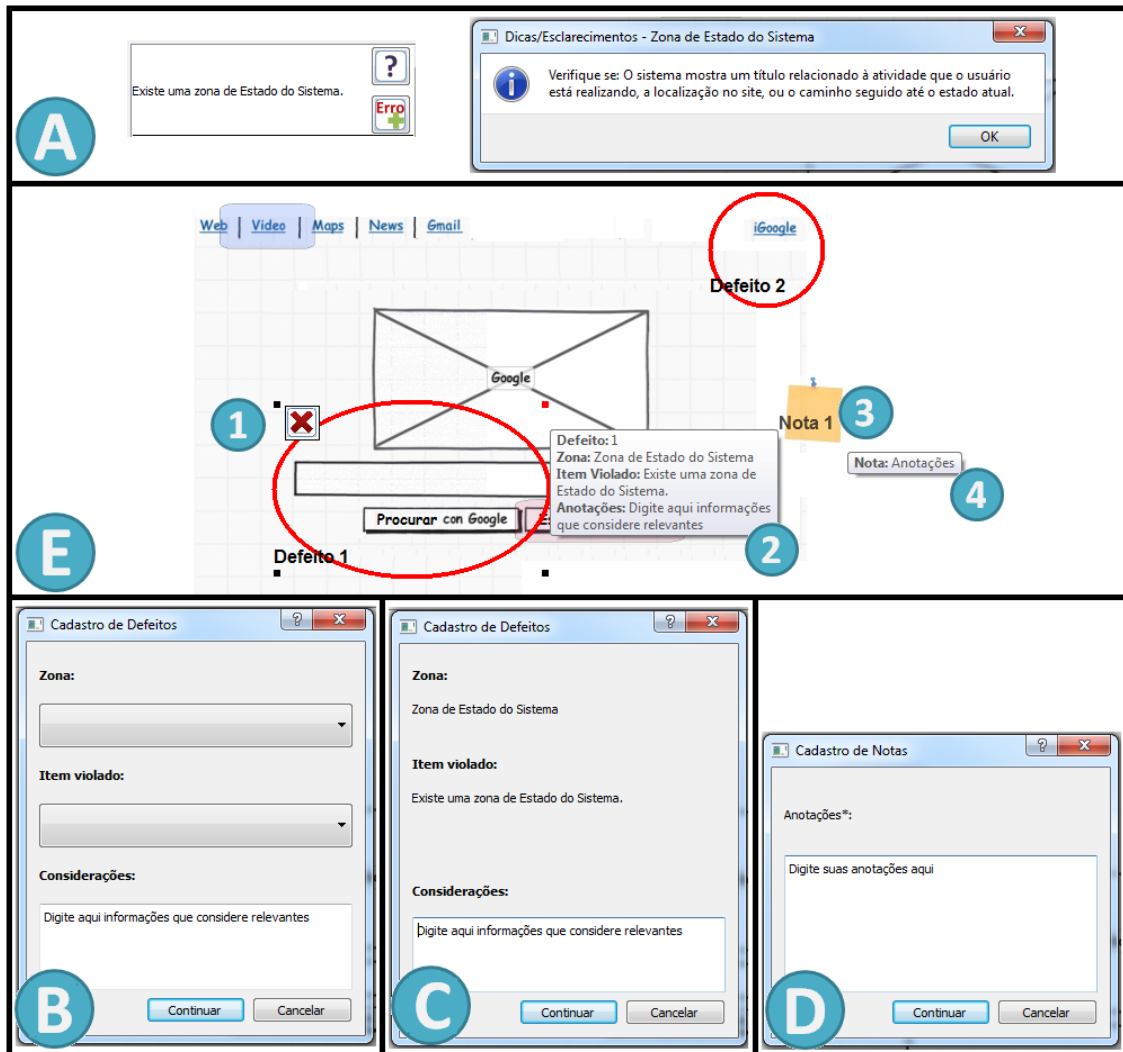


Figura 5-7 Funcionalidades do processo de detecção de problemas de usabilidade.

Para gerar um relatório de inspeção o inspetor deve clicar no botão nº 5 da Figura 5-6. A ferramenta *Mockup DUE* só consegue emitir um relatório quando a tela de detecção de defeitos está sendo mostrada. O relatório é gerado em um arquivo de formato *Portable Document Format (PDF)*. Este arquivo conterà todos os *mockups* e a localização dos problemas de usabilidade identificados. Também haverá uma tabela contendo os itens de verificação da técnica que foram violados, e as sugestões feitas pelo inspetor (caso hajam). A Figura 5-8 apresenta um exemplo básico do formato do relatório. Novamente, nesse exemplo os problemas apontados não são defeitos de usabilidade e servem somente para ilustrar o comportamento da ferramenta.

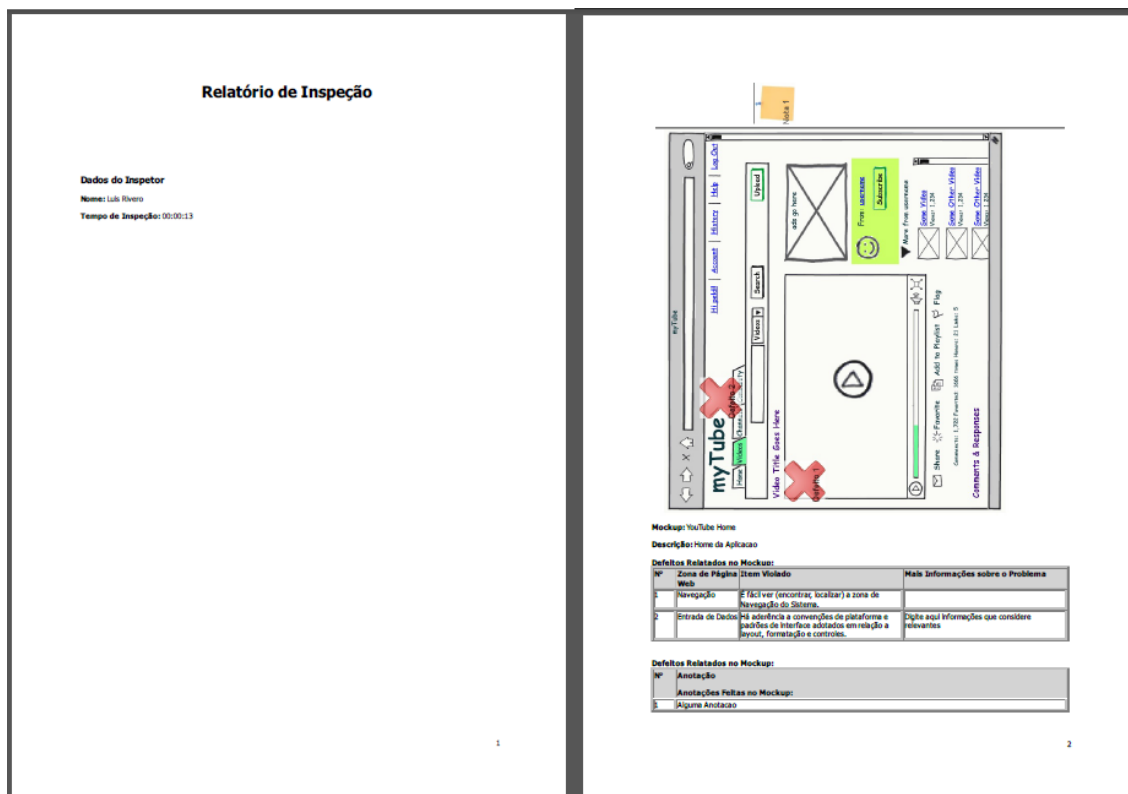


Figura 5-8 Exemplo de relatório gerado com a ferramenta *Mockup DUE*.

5.4 Considerações Finais

A análise dos resultados da extensão do mapeamento sistemático apresentado na Seção 2 apontou que as novas técnicas de inspeção de usabilidade de aplicações *Web* devem possuir apoio ferramental para simplificar seu processo de inspeção. Além disso, estas ferramentas devem ser permitir melhorar o desempenho da inspeção. Com o objetivo de suprir esta necessidade e melhorar o desempenho da técnica *Web DUE*, este capítulo apresentou a ferramenta *Mockup DUE*.

As funcionalidades da ferramenta *Mockup DUE* vieram dos resultados do estudo experimental da técnica *Web DUE* apresentado no Capítulo 4. Com o intuito de diminuir o esforço da inspeção, a ferramenta *Mockup DUE* automatiza o processo de mapeamento de *mockups* e simula a sua interação para que o inspetor possa se preocupar apenas na identificação de problemas de usabilidade. Além disso, este capítulo apresentou como a ferramenta permite apontar defeitos e fazer anotações que podem ser exportados em um relatório para posterior análise.

Para facilitar a identificação de problemas de usabilidade, a ferramenta *Mockup DUE* permite dois caminhos para cadastrar os defeitos, um para inspetores com pouca experiência no uso da técnica e da ferramenta e outro para inspetores mais experientes.

No seguinte capítulo, foi executado um estudo experimental para avaliar a viabilidade da ferramenta *Mockup DUE* em termos de satisfação do usuário. Os resultados apresentados no próximo capítulo permitem sugerir melhorias para facilitar o uso da ferramenta.

6 O Estudo de Viabilidade da Ferramenta *Mockup DUE*

Neste capítulo, é apresentado um estudo experimental para avaliar a viabilidade da Ferramenta Mockup DUE. Neste estudo, foi executada uma avaliação de usabilidade cooperativa com inspetores de média e alta experiência. Os dados qualitativos obtidos permitiram obter informações sobre a opinião dos inspetores com respeito ao seu grau de satisfação com a ferramenta, e à sua facilidade de uso.

6.1 Introdução

As novas técnicas de inspeção de usabilidade de aplicações *Web* devem possuir três características principais: (a) encontrar problemas de usabilidade nas primeiras etapas do processo de desenvolvimento; (b) encontrar problemas específicos de aplicações *Web*; e (c) possuir apoio ferramental para simplificar a sua aplicação. No Capítulo 5, com base nos resultados do estudo experimental da técnica *Web DUE*, foi proposta a ferramenta *Mockup DUE* que abstrai o processo de mapeamento de mockups do inspetor para que este possa focar sua atenção na identificação de problemas de usabilidade.

Para avaliar a ferramenta *Mockup DUE*, foi executado um estudo experimental para extrair informações sobre a sua facilidade de uso e o grau de satisfação dos seus usuários. Durante este estudo, inspetores de usabilidade com média e alta experiência realizaram uma avaliação cooperativa usando a ferramenta *Mockup DUE* para mapear os *mockups* de uma aplicação *Web*, e realizar uma inspeção de usabilidade. Com os dados extraídos, foi realizada uma análise qualitativa para identificar quão fácil era utilizar a ferramenta para executar o processo de inspeção da técnica *Web DUE*; e apontar o que pode ser modificado para melhorar o desempenho e o grau de satisfação dos seus usuários.

O restante deste capítulo foi dividido como segue. Na Seção 6.2 são apresentadas as etapas do estudo de viabilidade. A Seção 6.3 apresenta os resultados qualitativos, assim como a análise que permitiu sugerir modificações na ferramenta para facilitar o seu uso e melhorar o processo de inspeção. Na Seção 6.4, por sua vez, são apresentadas as ameaças à validade deste estudo experimental. Finalmente, na Seção 6.5, são apresentadas as considerações finais.

6.2 Estudo de Viabilidade

Neste capítulo foi planejado e executado um estudo de viabilidade para responder a seguinte questão de pesquisa: “É a ferramenta *Mockup DUE* viável em termos de satisfação do usuário e é fácil de usar?” Para responder a esta questão, foi feita uma avaliação cooperativa da ferramenta *Mockup DUE*. Em uma avaliação cooperativa, a equipe de design e os usuários trabalham juntos para avaliar um produto e identificar problemas de usabilidade e sugerir melhorias [24]. Nas seguintes subseções serão apresentadas as etapas deste estudo experimental, providenciando informações para avaliar a sua qualidade e a aplicabilidade dos seus resultados.

6.2.1 Objetivo

O objetivo deste estudo experimental foi avaliar a viabilidade da ferramenta *Mockup DUE* ao analisar os indicadores de satisfação do usuário e a facilidade de uso percebida pelos seus usuários. Para medir estes indicadores, foi analisada a opinião dos inspetores durante e após sua experiência com a ferramenta *Mockup DUE*. Além disso, foram identificadas as dificuldades que os inspetores tiveram ao mapear e visualizar *mockups*; e ao executar a detecção de problemas de usabilidade.

6.2.2 Participantes

O estudo de viabilidade foi executado em Junho de 2012 com inspetores de usabilidade experientes do Grupo USES da Universidade Federal do Amazonas. Houve um total de quatro alunos de pós-graduação que concordaram em participar. Todos os participantes assinaram um Termo de Consentimento Livre e Esclarecido e preencheram um formulário de caracterização. O formulário de caracterização avaliava a experiência dos participantes com respeito a: (a) conhecimento em usabilidade, (b) experiência em inspeções de usabilidade, e (c) experiência em design de aplicações. Todos os

participantes responderam a questões objetivas relacionadas à sua experiência profissional e grau de conhecimento. Os dados de caracterização foram analisados e cada participante foi classificado como tendo Baixa, Média ou Alta experiência, segundo as informações disponibilizadas no seu formulário de caracterização.

Os resultados da classificação dos participantes são mostrados na Tabela 6-1. Para classificar os inspetores foi utilizada a mesma classificação do primeiro estudo de viabilidade (ver subseção 4.2.3). Os resultados apresentados na Tabela 6-1 mostram que, de forma geral, os inspetores que participaram deste estudo possuíam de média a alta experiência em IHC e inspeções de usabilidade.

Além dos inspetores que utilizaram a ferramenta, houve outro papel importante na execução deste estudo de viabilidade. Um dos pesquisadores que desenvolveu a ferramenta Mockup DUE desempenhou o papel de moderador. O moderador era responsável por guiar a avaliação cooperativa e tomar nota de todos os problemas identificados com o uso da ferramenta, assim como sugestões de melhoria.

Tabela 6-1 Resultados da classificação dos participantes da avaliação cooperativa.

Experiência em:	Participante			
	1	2	3	4
Interação Humano Computador	Alta	Alta	Alta	Alta
Inspeções de Usabilidade	Média	Alta	Média	Alta
Design de Aplicações	Alta	Baixa	Média	Baixa

6.2.3 Materiais

Cada participante avaliou as principais características da ferramenta *Mockup DUE* usando dois conjuntos de *mockups*. O primeiro conjunto de *mockups*, que foi utilizado para testar as atividades de mapeamento e a simulação da interação entre *mockups*, foi baseado no site Google³. O outro conjunto de *mockups* foi baseado no site do sistema de submissão de artigos JEMS⁴. Este segundo conjunto foi previamente mapeado, já que os inspetores iriam utilizá-lo para realizar as atividades de detecção de defeitos de usabilidade fornecidas pela ferramenta *Mockup DUE*. Vale ressaltar que as aplicações *Web* usadas neste estudo experimental foram escolhidas de acordo com o grau de familiaridade dos inspetores com elas. O primeiro conjunto de *mockups* foi baseado no

³ <http://www.google.com.br/>

⁴ <https://submissoes.sbc.org.br/>

site Google, pois seria necessário que os inspetores tivessem experiência com a aplicação para que pudessem relacionar os *mockups*. O segundo conjunto de *mockups* foi baseado no sistema JEMS, pois já que os inspetores iriam realizar uma inspeção em cima destes *mockups*, não seria necessário nenhum conhecimento prévio sobre a interação com eles.

Para obter uma maior quantidade de dados qualitativos desta avaliação cooperativa, foi utilizado o software de teste de usabilidade Morae⁵. Este software permite ao pesquisador gravar e visualizar a reação do usuário com respeito ao software testado. Esta gravação pode ser utilizada para identificar passos de interação onde um problema pode ocorrer.

6.2.4 Procedimento

Todos os participantes receberam um conjunto de tarefas a serem realizadas usando a ferramenta *Mockup DUE*: (a) mapear a interação dos *mockups Web*; (b) simular a interação do usuário e o sistema em *mockups* de aplicações *Web*; e (c) detectar problemas de usabilidade. Os participantes foram informados de que o objetivo da avaliação cooperativa era a identificação de problemas de usabilidade relacionados à ferramenta *Mockup DUE*. Além disso, os participantes foram incentivados a sugerir melhorias para que a ferramenta pudesse ser usada mais facilmente. Durante a avaliação cooperativa, o moderador também fazia perguntas relacionadas à facilidade de uso da ferramenta *Mockup DUE*. Se o participante encontrava alguma dificuldade ou facilidade ao utilizar a ferramenta, o moderador anotava estes pontos e identificava as causas. Adicionalmente, o software Morae foi utilizado para gravar as seções com os participantes para posteriores análises. A Figura 6-1 apresenta um participante realizando a avaliação cooperativa da ferramenta *Mockup DUE* capturado utilizando o software de teste Morae.

⁵ <http://www.techsmith.com/morae.html>

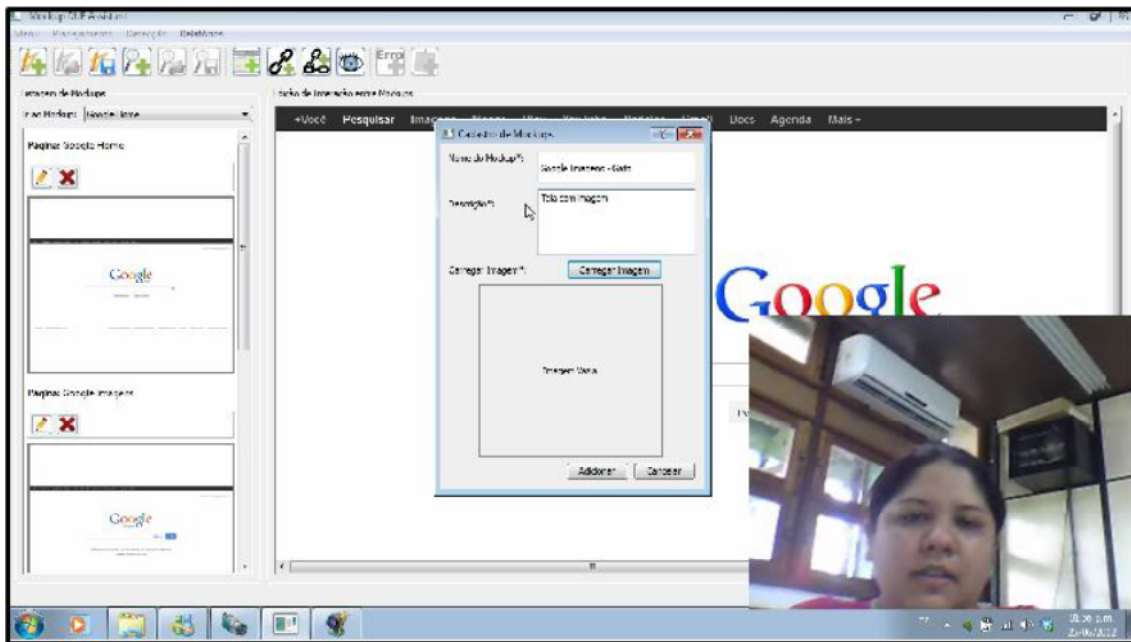


Figura 6-1 Participante aplicando a ferramenta *Mockup DUE*.

Cada participante teve tanto tempo quanto necessário para realizar as atividades relacionadas à ferramenta *Mockup DUE*. Além disso, os participantes não realizaram uma inspeção completa do segundo conjunto de *mockups* disponibilizado, já que o foco do estudo era a avaliação da ferramenta *Mockup DUE* como assistente ao processo de inspeção da técnica *Web DUE*. Como consequência, não houve necessidade de realizar uma inspeção completa. Finalmente, no final da avaliação cooperativa, os participantes preencheram um questionário com comentários relacionados à ferramenta *Mockup DUE*.

6.2.5 Coleção de Dados

Foram coletados três tipos de informações: (a) anotações do moderador durante a avaliação cooperativa; (b) os questionários pós-inspeção contendo comentários dos inspetores e seu grau geral de satisfação; e (c) os vídeos das avaliações com cada um dos participantes. Na próxima seção serão apresentados os resultados qualitativos deste estudo de viabilidade e a sua análise.

6.3 Análise Qualitativa

A análise dos resultados começou pela avaliação das respostas contidas nos questionários de pós-inspeção. Neste questionário, foi solicitado aos participantes que indicassem seu grau geral de satisfação com a ferramenta usando uma Escala Análoga

Visual (Visual Analogue Scale - VAS) [33]. Uma VAS é um instrumento de medida que serve para mensurar as características ou atitudes que se acreditam estar em um intervalo contínuo de valores, e que não podem ser diretamente mensurados [33]. A Figura 6-2 apresenta os resultados gerais de satisfação dos participantes ao utilizarem a ferramenta *Mockup DUE*. Estes resultados serão relacionados às observações feitas durante a avaliação cooperativa e às respostas dos participantes ao questionário pós-inspeção.

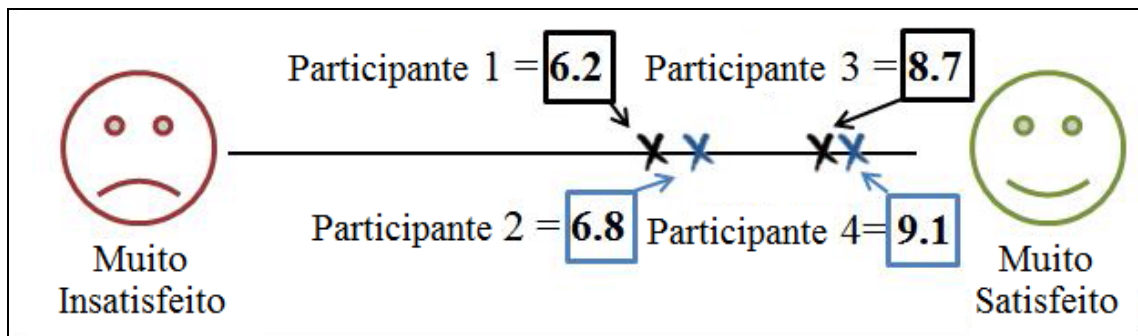


Figura 6-2 Resultados gerais de satisfação dos participantes.

Para identificar que fatores afetaram a opinião dos inspetores com relação à ferramenta, também foram feitas as perguntas da Figura 6-3.

1. Do ponto de vista de inspeção de usabilidade, qual sua opinião em relação à ferramenta Mockup DUE Assistant sobre **facilidade de uso**? Comente.
2. Do ponto de vista de inspeção de usabilidade, qual sua opinião em relação à **adequação** da ferramenta Mockup DUE Assistant para **inspeção de mockups**? Comente.
3. As atividades das etapas (Planejamento e Detecção) apoiadas pela ferramenta Mockup DUE Assistant foram disponibilizadas de forma que **fosse possível executá-las de forma fácil sem treinamento prévio**? Em caso negativo, o que poderia ser melhorado no design de Interação ou processo seguido pela ferramenta para executar as atividades?
4. Das atividades/opções/instruções disponibilizados pela ferramenta, você considera que foram **intuitivas e fáceis de executar**? Em caso negativo, o que poderia ser melhorado para facilitar seu entendimento e execução?
5. Dos ícones/imagens/layout disponibilizados pela ferramenta, existe algum que você não tenha entendido ou que tenha atrapalhado o processo de planejamento ou detecção de defeitos? Em caso afirmativo, quais foram e que poderia ser mudado para melhorar o **design da Ferramenta**?
6. Quais **aspectos** você considera que fizeram a ferramenta **fácil ou difícil de usar**? Comente.
7. Se você tivesse que realizar uma nova avaliação de usabilidade de mockups, seria possível **considerar a utilização dessa Ferramenta**? O que você sugeriria nesta reaplicação da ferramenta?

Figura 6-3 Questões relacionadas à facilidade de uso da ferramenta.

Os resultados para a pergunta Q1 (É a ferramenta fácil de usar?) e a Q6 (Quais são os aspectos que fazem a ferramenta fácil ou difícil de usar?) mostraram que a opinião dos inspetores estava dividida. Os inspetores 1 e 2 acharam a ferramenta difícil de usar, pois acharam que era necessária mais informação para usá-la. Já os inspetores 3 e 4, acharam que, no geral, a ferramenta era fácil de usar, mas que era necessário realizar algumas mudanças para melhorar a sua facilidade de uso. O inspetor 3 afirmou que quando os itens de verificação eram mostrados, a ferramenta omitia algumas palavras por falta de espaço. O inspetor 4 indicou que mais informação era necessária para entender qual era o estado do sistema com relação ao processo de inspeção. As citações dos participantes abaixo sustentam estas afirmações.

“A ferramenta precisa melhorar sua usabilidade. Eu me perdi quando comecei a mapear os mockups e comecei a inspeção.”

- Inspetor 1.

“A ferramenta devia melhorar alguns dos elementos da interface. Existem algumas ambiguidades com as atividades de inspeção e as atividades do usuário.”

- Inspetor 2.

“A ferramenta é fácil de usar pois os ícones são intuitivos... No entanto, quando adicionei um erro de usabilidade, não conseguia ler a descrição.”

- Inspetor 3.

“É muito legal, bonita Mas tem uns detalhes que a fazem confusa... Ao trabalhar com os links não tem informação sobre qual mockup eu deixei...”

- Inspetor 4.

As respostas para a pergunta Q2 (A ferramenta é adequada para a inspeção de *Mockups Web*?) indicaram que todos os inspetores concordaram que era muito adequada (ver citação do inspetor 3). No entanto, o inspetor 4 afirmou que não ficou claro se era possível ou não apontar um problema de usabilidade que não estivesse anotado na lista de itens de verificação de usabilidade.

“Eu gostei muito, e com certeza foi muito melhor que usar os mockups Web...”

- Inspetor 3.

“É adequada, porém, existem alguns detalhes que confundem o usuário, por exemplo: eu pensei que só podia apontar os problemas da lista.”

- Inspetor 4.

Os resultados para a pergunta Q3 (É necessário treinamento antes de usar a ferramenta?) indicaram que alguns inspetores precisavam de mais informação antes de

usar a ferramenta (ver citações dos inspetores 1 e 2). No entanto, os inspetores 3 e 4 afirmaram que as atividades eram fáceis de executar sem treinamento prévio. Ainda assim, esses inspetores disseram que era necessário melhorar alguns problemas de usabilidade (ver citação do inspetor 4).

“Não, algumas dicas deviam ser disponibilizadas ao começar as atividades.”

- Inspetor 1.

“Sem a influência do observador pode ser muito difícil usar a ferramenta sem treinamento. Alguns ícones, funcionalidades e módulos possuem nomes inadequados. ... Algumas funcionalidades podem ser agrupadas de forma diferente para melhorar a experiência de usuário.”

- Inspetor 2.

“Sim, apesar de ser um pouco confuso e difícil de aprender, é possível usar sem treinamento. Talvez podia haver uma descrição geral das funcionalidades...”

- Inspetor 4.

Com respeito à questão Q4 (As funcionalidades são intuitivas e fáceis de realizar?) e a questão Q5 (O layout da aplicação é fácil de entender?), foi identificado que o layout tem um efeito direto sobre quão fácil era executar as funcionalidades. Os inspetores afirmaram que, no geral, foram capazes de realizar todas as atividades. No entanto, a localização, tamanho e descrição dos elementos da interface dificultava este processo. As citações dos inspetores 1 e 4 ilustram este fato.

“Alguns elementos da interface devem ser melhorados... Na fase de detecção os itens devem ser melhor organizados, assim como os itens de verificação e os botões.”

- Inspetor 1.

“Eu entendi tudo com o que tive contato. Mas, tive que ler as dicas para entender a funcionalidade dos botões... Acredito que alguns dos elementos da interface não foram bem localizados.”

- Inspetor 4.

Finalmente, as respostas à questão Q7 (É possível considerar a utilização da ferramenta na inspeção de *Mockups Web*?) mostrou que todos os inspetores utilizariam a ferramenta para realizar inspeções de *mockups* de aplicações *Web*. No entanto, a maioria dos inspetores sugeriu a melhoria da interação. Para isso, os inspetores citaram a correção dos problemas citados acima (ver citações dos inspetores 2 e 3).

“Sim, mas melhorar as palavras termos, e alguns itens e sua localização.”

- Inspetor 2.

“Claro que a usaria de novo, se os erros anteriores fossem corrigidos.”

- Inspetor 3.

No geral, a média de satisfação dos inspetores com relação ao uso da ferramenta foi de 7.7 (com base na escala VAS da Figura 6-2). Isto significa que os inspetores estavam satisfeitos ao utilizar a ferramenta. Foi identificado que existe uma relação entre a facilidade de uso percebida e o grau de satisfação dos inspetores. Os inspetores 3 e 4 avaliaram a ferramenta *Mockup DUE* com um alto grau de satisfação. Esta avaliação se deve ao fato que estes inspetores acharam a ferramenta *Mockup DUE* fácil de usar desde o início e também não precisaram de treinamento para usá-la. Por outro lado, os inspetores 1 e 2 afirmaram que para eles a ferramenta não oferecia informações suficientes, o que afetou o seu grau de satisfação.

Para identificar que elementos da interface ocasionaram um efeito negativo sobre a opinião dos inspetores com respeito à ferramenta, foram analisadas as gravações e anotações feitas durante a avaliação cooperativa. Na Tabela 6-2 são listadas as correções que foram feitas para melhorar a usabilidade da ferramenta *Mockup DUE*. Estas modificações serão incorporadas em novas versões da ferramenta. Na próxima seção, serão discutidas as ameaças à validade deste estudo de viabilidade da ferramenta *Mockup DUE*.

Tabela 6-2 Mudanças sugeridas pelos inspetores durante a avaliação cooperativa.

Elemento	Problema	Solução
Atividade de Mapeamento de <i>Mockups</i> – Listagem de <i>Mockups</i> : Botão de Editar.	O nome do botão sugere ao usuário que editará o <i>mockup</i> , quando o que será editado é a interação.	Mudar o nome do botão para: “Editar Mapeamento”.
Atividade de Mapeamento – Links: Botão de Deletar.	A ferramenta não pode confirmação antes de apagar um link. Se o usuário clica no botão por erro, não será capaz de desfazer a ação.	Solicitar confirmação antes de apagar um link.
Atividade de Mapeamento – Toda a interface.	O usuário não sabe em que parte do sistema se encontra.	Adicionar informações sobre o estado do sistema.
Atividade de Detecção: Notas.	O usuário só pode ver as anotações quando passa o mouse em cima dela e	Mostrar a dica, assim que o usuário passar o mouse em cima da nota.

Elemento	Problema	Solução
	espera que apareça o <i>tooltip</i> .	
Atividade de Detecção – Toda a interface.	O usuário não sabe se um <i>mockup</i> foi corretamente carregado e se foi mostrado.	Quando um <i>mockup</i> for mostrado, fornecer dicas de que um novo <i>mockup</i> foi carregado, por meio de um som, ou simular que a <i>mockup</i> está carregando.
Atividade de Detecção – Apontador de defeitos.	Quando o usuário apontar um defeito é melhor utilizar um “X” a um círculo.	Substituir o símbolo que representa um problema de usabilidade.

6.4 Ameaças à Validade

Todo estudo experimental possui ameaças que podem afetar a validade dos seus resultados. Nesta seção, são apresentadas as ameaças à validade deste estudo e como foram tomadas medidas para mitigá-las.

Foram consideradas cinco ameaças à validade: (a) estudantes não são bons substitutos de inspetores profissionais, (b) ambientes acadêmicos não representam a experiência do dia a dia na indústria, (c) se os sites Google e JEMS são aplicações *Web* representativas de todas as aplicações *Web*, (d) o número de participantes envolvidos no estudo, e (e) se os critérios escolhidos para medir a viabilidade da ferramenta foram escolhidos corretamente.

Com relação à primeira ameaça, o uso de estudantes como inspetores, vale ressaltar que uma vez que os participantes possuíam altos níveis de conhecimento de usabilidade, e experiência em usabilidade, estes podiam ser considerados inspetores experientes. Além disso, apesar de ter utilizado um ambiente acadêmico para executar o estudo de viabilidade, os *mockups* utilizados foram baseados em aplicações *Web* reais, o que pode ajudar a simular um ambiente real de desenvolvimento de software (ameaça b). Com respeito ao uso dos sites Google e JEMS, não pode ser afirmado que esta não seja uma ameaça, visto que existem muitas categorias de aplicações *Web* [18].

O maior problema neste estudo é o tamanho da amostra. Como só houve quatro inspetores que participaram da avaliação cooperativa, os dados extraídos deste estudo experimental devem ser considerados indicativos e não conclusivos. Finalmente, a satisfação e facilidade de uso percebida são critérios comuns para a investigação da

produtividade de novos softwares [37]. Consequentemente, a ameaça (e) não pode ser considerada um risco para a validade dos resultados obtidos.

6.5 Considerações Finais

A ferramenta *Mockup Design Usability Evaluation (Mockup DUE)* tem por objetivo apoiar o processo de inspeção de *mockups* de aplicações *Web*. A principal característica desta ferramenta é sua habilidade de mapear a interação entre os *mockups*, e ajudar os inspetores a realizar uma inspeção de usabilidade utilizando a técnica *Web DUE*.

Neste capítulo, foi apresentado um estudo de viabilidade para responder se a ferramenta *Mockup DUE* é viável em termos de grau de satisfação dos usuários, e facilidade de uso percebida. Para responder a esta pergunta foi executada uma avaliação cooperativa usando inspetores com média e alta experiência em avaliações de usabilidade. Os resultados qualitativos deste estudo ofereceram indícios da viabilidade da ferramenta para a inspeção de *mockups*, porém também forneceram sugestões de melhoria para as próximas versões da ferramenta.

Foi identificado que a *Mockup DUE* é considerada adequada para os processos de mapeamento e inspeção de *mockups* de aplicações *Web*. Além disso, a média do grau de satisfação dos usuários foi de 7.7 usando uma escala análoga visual. Os inspetores também afirmaram que utilizariam a ferramenta *Mockup DUE* para realizar inspeções de usabilidade no contexto de *mockups* de aplicações *Web*. Adicionalmente, foi identificado que, para ser usada por inspetores novatos no uso da técnica *Web DUE*, deve ser providenciado um treinamento sobre a ferramenta, ou incluir um manual que descreva as atividades a serem realizadas no processo de inspeção da técnica *Web DUE*. Os inspetores também sugeriram mudanças no layout, localização, tamanho e descrição de alguns dos elementos da interface da ferramenta para facilitar o seu uso.

No seguinte capítulo, são apresentadas as modificações a serem feitas em cima da técnica *Web DUE* e da ferramenta *Mockup DUE*. Com base nos resultados dos estudos experimentais realizados, serão propostas as novas versões de cada uma dessas propostas de inspeção de usabilidade de *mockups* de aplicações *Web*.

7 A Evolução da Técnica *Web DUE* e da Ferramenta *Mockup DUE*

Os resultados de ambos os estudos de viabilidade executados apontam que existem pontos a serem melhorados tanto na técnica quanto na ferramenta. Neste capítulo são apresentadas as novas versões da técnica e da ferramenta, que foram desenvolvidas integrando as mudanças sugeridas pelos participantes dos estudos e as análises dos resultados.

7.1 Introdução

Os estudos de viabilidade da técnica *Web DUE* e da ferramenta *Mockup DUE* permitiram obter indícios da sua viabilidade. No entanto, estes também apontaram oportunidades de melhoria. Neste capítulo, é proposta a segunda versão da técnica *Web DUE*. Foi feita uma análise técnica dos itens de verificação da técnica *Web DUE* para encontrar quais deles podiam levar os inspetores a apontarem falsos positivos. Já na segunda versão da ferramenta *Mockup DUE*, foram integradas as sugestões no layout dos elementos da ferramenta, assim como a criação de um banco de dados para armazenar os dados da inspeção. Além destas mudanças, foi proposta o design de uma nova versão da ferramenta *Mockup DUE*, que guia os inspetores pelo processo de inspeção da técnica *Web DUE*.

Este capítulo apresenta as análises decorrentes dos estudos de viabilidade que levaram a propor as novas versões da técnica *Web DUE* e da ferramenta *Mockup DUE*. A Seção 7.2 apresenta as mudanças sugeridas a partir do primeiro estudo de viabilidade da técnica *Web DUE* e que foram integradas à técnica. Na Seção 7.3 é apresentada a nova versão da ferramenta *Mockup DUE*, onde foram feitas mudanças no design da ferramenta e a integração com um banco de dados. A Seção 7.4 apresenta o design de uma nova versão da ferramenta *Mockup DUE* que tem por objetivo guiar os inspetores novatos pelo processo de inspeção da técnica *Web DUE*, utilizando a ferramenta. Finalmente a Seção 7.5 apresenta as considerações finais deste capítulo.

7.2 A Técnica *Web DUE* v2

Os resultados qualitativos do Capítulo 4 deram indícios da viabilidade da técnica *Web DUE* para a inspeção de *mockups* de aplicações *Web*. Os inspetores consideraram a técnica adequada e afirmaram que o uso de zonas próprias de páginas *Web* podia ajudar os inspetores a encontrar mais problemas de usabilidade, pois estes focavam em partes específicas do *mockup*, o que facilitava a identificação de problemas. No entanto, alguns dos itens de verificação disponibilizados pela técnica *Web DUE* foram considerados confusos, como pode ser observado na seguinte citação:

“... No entanto, alguns itens de verificação julgam atributos que podem não estar totalmente certos ou errados.”

- Inspetor 3 que participou do estudo de viabilidade da técnica *Web DUE*.

Apesar de solicitar aos inspetores que indicassem quais itens de verificação eram confusos ou difíceis de entender, nenhum inspetor deu mais detalhes. Por esse motivo, e uma vez que outros participantes apontaram o mesmo problema com relação aos itens de verificação da técnica, foi realizada uma análise mais detalhada dos resultados obtidos no estudo de viabilidade. Durante esta análise, buscou-se identificar quais dos itens de verificação da técnica *Web DUE*, podiam levar a identificação errônea de problemas de usabilidade. Em outras palavras, buscou-se identificar que itens de verificação levaram os inspetores a apontar falsos positivos (discrepâncias que, na realidade, não são problemas de usabilidade).

A Tabela 7-1 apresenta a listagem de todos os itens de verificação que foram marcados como violados pelos inspetores e que, no entanto, foram considerados como falsos positivos durante a reunião de discriminação. A partir da Tabela 7-1 é possível identificar dois fatores que contribuíram para a identificação dos falsos positivos: (a) o treinamento levou os inspetores a apontar todos os itens que foram violados, mesmo que os inspetores não os considerassem problemas reais de usabilidade; e (b) os exemplos/esclarecimentos eram considerados, pelos inspetores novatos, como a forma de verificar se o item tinha sido violado ou não.

O fato de apontar problemas que realmente não afetam a usabilidade do sistema está relacionado ao treinamento da técnica *Web DUE*. Segundo as instruções da técnica *Web DUE*, um inspetor deverá reportar qualquer não conformidade com os itens de verificação como um problema de usabilidade (ver instruções da técnica *Web DUE* v1 no Apêndice B). Essa instrução foi reforçada no treinamento ministrado durante o

estudo de viabilidade descrito no Capítulo 4. Consequentemente, os inspetores apontaram os itens de verificação da Tabela 7-1 em negrito (Itens 02, 04, 05, 07, 08, 09, 10, 14 e 16) como sendo defeitos de usabilidade. No entanto, uma análise cuidadosa das descrições dos defeitos permitiu identificar que os inspetores apontaram esses defeitos como “não problemas” ou “cosméticos”. Segundo Nielsen [26], defeitos classificados como “não problemas” são aqueles que o inspetor não considera defeitos de usabilidade, enquanto defeitos classificados como “cosméticos” são aqueles problemas de usabilidade com mínima prioridade que só precisam ser corrigidos se sobrar tempo no projeto de desenvolvimento. Ou seja, os inspetores só apontaram esses falsos positivos por causa das instruções da técnica *Web DUE*. Para evitar que os inspetores apontem falsos positivos por causa de uma não conformidade optou-se por modificar a instrução da técnica *Web DUE* como segue:

“Se o mockup não estiver aderente ao item de verificação, avalie se a não conformidade pode ocasionar um problema de usabilidade na aplicação Web.”

Com essa instrução espera-se aumentar a adequação dos defeitos de usabilidade encontrados, diminuindo o número de falsos positivos e apoiando na identificação de problemas de usabilidade que possam atrapalhar a realização das tarefas do sistema.

Tabela 7-1 Itens de verificação que levaram os inspetores a apontar falsos positivos.

Nº	Zona	Item de Verificação	Exemplo/Esclarecimento
01	<i>Customizável</i>	<i>a) O usuário conseguirá identificar que errou.</i>	
02	Entrada de Dados	A interface do sistema permite ao usuário visualizar informações sobre o formato correto de entrada de dados.	As dicas de preenchimento devem ser facilmente visualizadas pelo usuário.
03	<i>Entrada de Dados</i>	<i>A interface facilita a distinção entre dados diferentes.</i>	<i>Em caso de serem solicitadas dois dados similares, cada um deve ser identificado. Por exemplo: Data de vencimento, data de pagamento.</i>
04	Entrada de Dados	A interface permite ao usuário a utilização de aceleradores ou atalhos na inserção de dados.	Por exemplo, um usuário pode salvar os dados do seu cartão de crédito para não preencher os mesmos dados a cada compra.
05	Entrada de Dados	Avalie se a interface permite a utilização de mecanismos de busca de informação que auxiliem a	Por exemplo, na hora de inserir um endereço, o sistema pode permitir procurar endereços no

Nº	Zona	Item de Verificação	Exemplo/Esclarecimento
		entrada de dados obrigatória.	google maps.
06	<i>Entrada de Dados</i>	<i>Existe uma zona de Entrada de Dados no Sistema.</i>	<i>O sistema apresenta meios de inserção de dados: formulários, opções para selecionar itens, opções para marcar itens, ou meios de fornecer dados ao sistema a fim de realizar uma tarefa.</i>
07	Entrada de Dados	Tarefas similares possuem formulários de preenchimento similares. O estilo de preenchimento deve ser padronizado.	Ao informar dados em tarefas similares, os formulários não podem mudar radicalmente. Isso permite ao usuário saber como preencher dados em determinadas tarefas.
08	Estado do Sistema	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
09	Informação	A interface permite ao usuário diferentes formas de acesso às informações.	As informações podem ser visualizadas seguindo caminhos diferentes.
10	Informação do Usuário	As informações do usuário são apresentadas de forma ordenada e lógica.	Por exemplo, num site de compras, os dados relevantes do usuário podem ser mostrados na ordem: E-mail, saldo disponível, pontos promocionais, etc.
11	<i>Navegação</i>	<i>a) O sistema mostra uma mensagem de erro.</i>	<i>Se uma página não está disponível o sistema mostra uma mensagem.</i>
12	<u>Navegação</u>	<u>É fácil entender que essa é a zona de navegação do Sistema.</u>	<u>É fácil reconhecer que o menu de navegação permite o acesso a outras partes do sistema.</u>
13	<u>Navegação</u>	<u>Existe uma zona de Navegação no Sistema.</u>	<u>O sistema possui um menu de navegação.</u>
14	Serviços	A zona de Serviços é padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Ao usar um símbolo (como uma lupa para o serviço de busca) o mesmo deve ser usado ao longo da aplicação.
15	<i>Serviços</i>	<i>a) O sistema mostra uma mensagem de erro.</i>	<i>Se um serviço não está disponível o sistema mostra uma mensagem.</i>
16	Serviços	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Por exemplo, o símbolo do botão "busca" é representado por uma lupa.

Em relação aos exemplos/esclarecimentos dos itens de verificação, foi observado que os inspetores os usavam como referência para tomar decisões sobre se incluir ou não um problema de usabilidade. Na Tabela 7-1, os itens de verificação em itálico (Itens 01, 03, 06, 11 e 15), e os itens de verificação sublinhados (Itens 12 e 13)

são falsos positivos que foram apontados, pois o *mockup* não satisfazia as pré-condições para sua avaliação ou por falta de informações relevantes.

O primeiro grupo de itens de verificação (em itálico) foi considerado como problemas de usabilidade, pois os *mockups* não estavam aderentes a eles. No entanto, uma análise mais detalhada revelou que a verificação desses itens depende da funcionalidade do *mockup*. A Figura 7-1 apresenta um exemplo em que para avaliar se o *mockup* se adere ao item de verificação 03 “*A interface facilita a distinção entre dados diferentes.*”, é necessário que a zona de entrada de dados solicite dados diferentes do mesmo tipo no mesmo formulário de entrada de dados (ver Figura 7-1 parte A). Do contrário, se não houver dados diferentes, o sistema não terá porque fazer distinção entre eles (ver Figura 7-1 parte B) e, como consequência, o inspetor relatará a não conformidade como um problema de usabilidade. Para evitar problemas desse tipo, itens de verificação que possuem uma pré-condição para serem avaliados, foram renomeados na segunda versão da técnica *Web DUE* como mostra a Figura 7-2. Além disso, foi inserida mais uma instrução do processo de detecção de defeitos da técnica *Web DUE* para verificar se o item pode ser avaliado:

“Avalie se o item de verificação se aplica ao mockup que está sendo inspecionado. Isto é, as pré-condições necessárias para avaliar o item estão sendo respeitadas pelo mockup.”

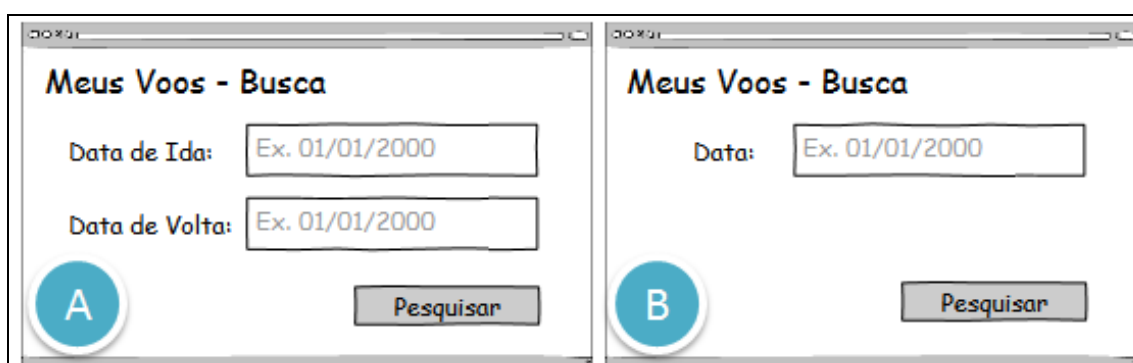


Figura 7-1 Exemplo onde os itens de verificação da técnica Web DUE podem e não ser avaliados.

Zona de Entrada de Dados	
Técnica Web DUE v1	Técnica Web DUE v2
A interface facilita a distinção entre dados diferentes.	Se são solicitados dados diferentes: A interface facilita a distinção entre os dados.

Figura 7-2 Exemplo de item renomeado para evitar a identificação de falsos positivos.

O segundo grupo de itens de verificação (sublinhado) foi considerado de falsos positivos, pois os exemplos/esclarecimentos não continham informações suficientes para apoiar os inspetores na tomada de decisões. Por exemplo, o item de verificação 13 “Existe uma zona de Navegação no Sistema” está relacionado com o exemplo/esclarecimento “O sistema possui um menu de navegação”. No entanto, um inspetor novato ao ler o exemplo, pode considerar que as zonas de navegação são apenas os menus e não qualquer outro meio de navegação na página. Para evitar este tipo de problema reescreveram-se os exemplos/esclarecimentos para providenciar a maior quantidade de informação e da forma mais clara possível. A Figura 7-3 apresenta o exemplo/esclarecimento reescrito para o item de verificação 13 da Tabela 7-1.

Pode-se notar que nem todos os itens de verificação ou seus exemplos/esclarecimentos foram renomeados, uma vez que só foram relatados 16 itens de verificação cujas descrições podiam levar a apontar falsos positivos. No entanto, foram revisados os itens de verificação similares para corrigir os problemas apontados no primeiro estudo de viabilidade. A versão completa da técnica *Web DUE* v2 pode ser visualizada no Apêndice C.

Zona de Navegação	
Técnica Web DUE v1	Técnica Web DUE v2
O sistema possui um menu de navegação.	O sistema possui um conjunto de links de navegação , que podem ser ativados para se deslocar de uma página a outra.

Figura 7-3 Exemplo de exemplo/esclarecimento reescrito na técnica *Web DUE* v2.

7.3 A Ferramenta *Mockup DUE* v2

Como sugerido pelos inspetores que participaram da avaliação cooperativa relatada no Capítulo 6, a ferramenta *Mockup DUE* sofreu alterações para facilitar o seu entendimento e melhorar a sua usabilidade. Entre as modificações feitas com base nas sugestões da Tabela 6-2. A seguir apresentaremos imagens da nova versão da ferramenta *Mockup DUE* e suas alterações.

A Figura 7-4 mostra como é que o estado do sistema foi apresentado aos usuários. Na versão da ferramenta *Mockup DUE*, a barra de título apresenta o arquivo que está sendo editado e em que tipo de atividade o inspetor está trabalhando, seja esta o planejamento ou a detecção de defeitos de usabilidade. Além disso, o sistema informa ao usuário quais atividades estão sendo realizadas durante a execução de uma tarefa automatizada. As tarefas automatizadas consistem em salvar e abrir planejamentos; salvar e abrir processos de detecção de defeitos de usabilidade; ou gerar um relatório de inspeção.

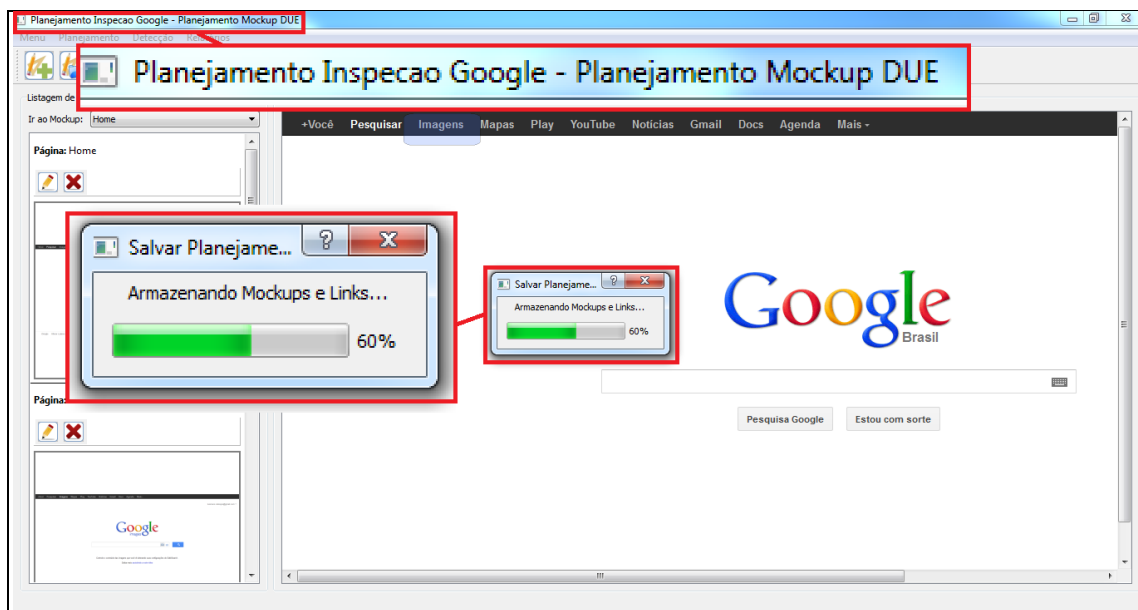


Figura 7-4 Informando o estado do sistema na ferramenta *Mockup DUE* v2.

Na Figura 7-5, o botão “*Editar Mockup*” foi renomeado para “*Editar Mapeamento do Mockup*”. Essa modificação foi feita para não confundir o inspetor e deixar claro que as ações de edição estão relacionadas ao mapeamento dos *mockups*. Além disso, a Figura 7-6 apresenta a nova versão da ferramenta solicitando confirmação

antes de realizar ações que não possam ser desfeitas, como apagar um link, defeito, ou nota.

Com relação à atividade de detecção de defeitos, a Figura 7-7 apresenta três modificações na nova versão da ferramenta *Mockup DUE*: (a) mudança na imagem para apontar defeitos, (b) tempo de resposta para mostrar notas, Com relação à mudança da imagem para apontar defeitos, os inspetores sugeriram utilizar um “X”, visto que um círculo podia não ser percebido como um problema de usabilidade.

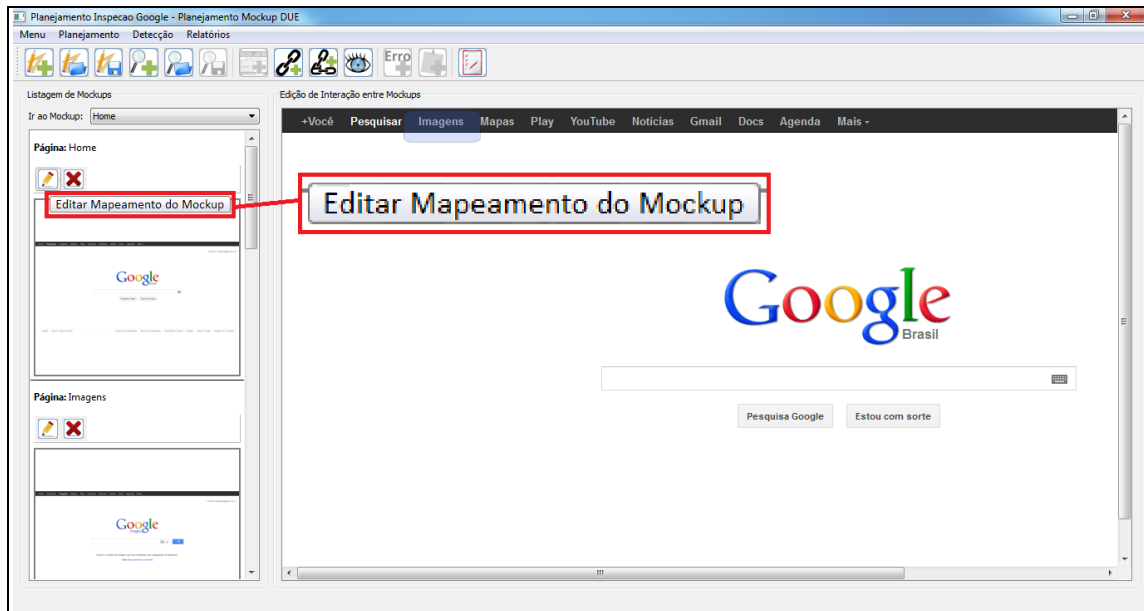


Figura 7-5 Renomeando botões para facilitar o entendimento na *Mockup DUE* v2.

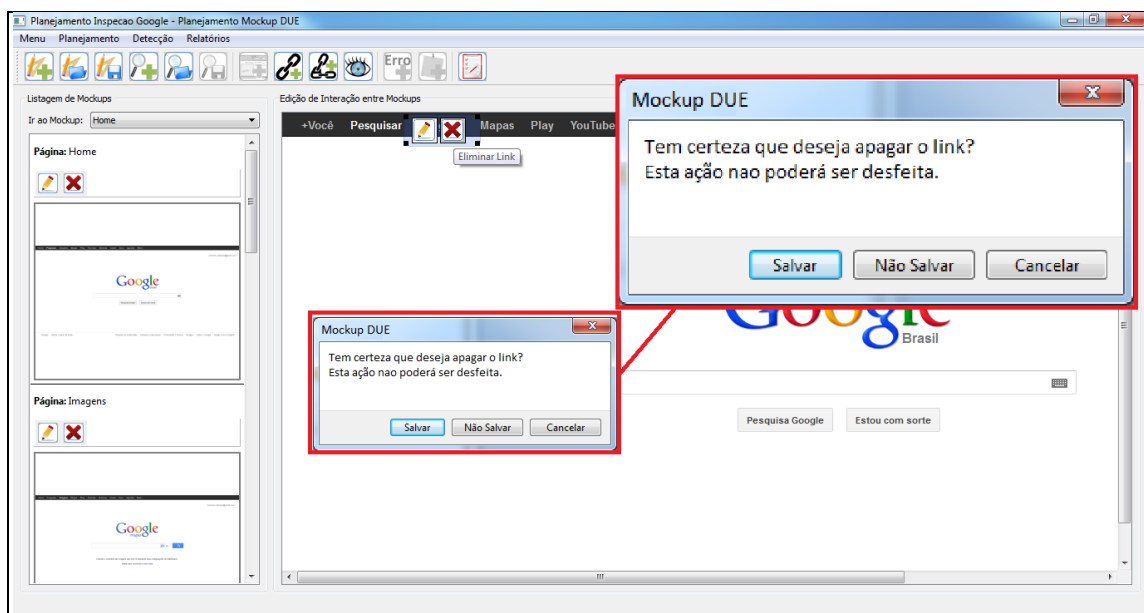


Figura 7-6 Solicitando confirmação na ferramenta *Mockup DUE* v2.

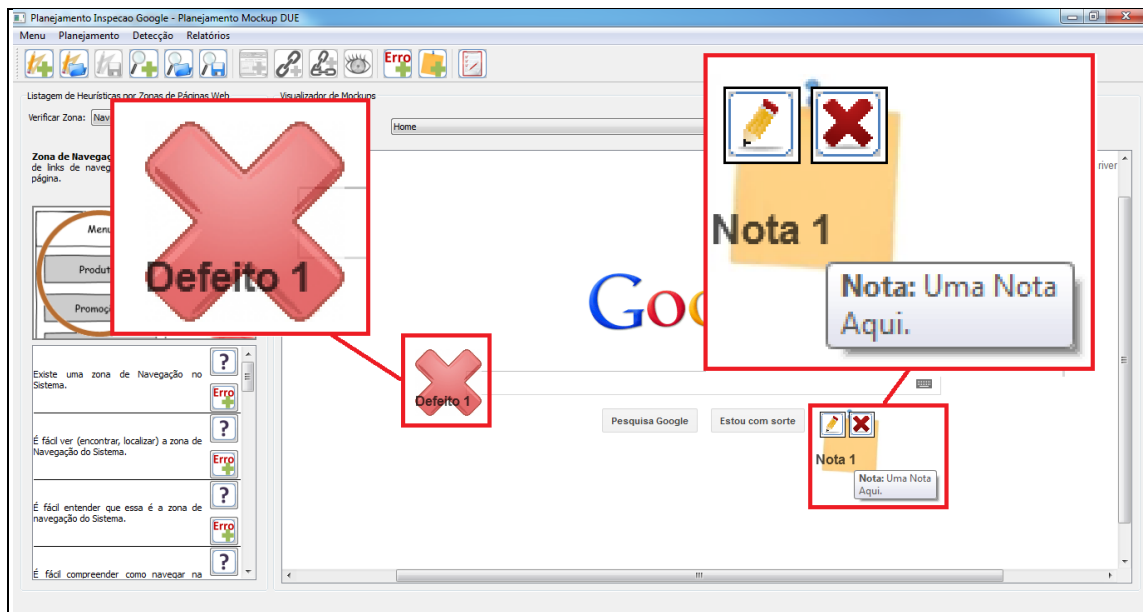


Figura 7-7 Modificações na atividade de detecção de defeitos.

Para melhorar o tempo de resposta da ferramenta em apresentar informações importantes, como as anotações e/ou as descrições dos defeitos, itens violados e zonas de páginas *Web* onde ocorreram os defeitos, optou-se por apresentar estas informações assim que o usuário passasse o mouse em cima das notas e defeitos apontados. Esta mudança facilita a visualização da informação durante o processo de inspeção. Além disso, a mesma medida foi tomada para fornecer informações sobre os links. Consequentemente, assim que o usuário passa o mouse em cima de um link, o sistema indica qual é o seu destino.

Finalmente, para proporcionar informações sobre o carregamento de um *mockup*, seja durante o mapeamento ou detecção de problemas, a ferramenta *Mockup DUE* toca um som de “beep” para sinalizar o carregamento.

Outras mudanças foram incorporadas à ferramenta *Mockup DUE*, principalmente no que diz respeito ao armazenamento das informações. Na primeira versão da ferramenta, os dados do planejamento e detecção de defeitos eram armazenados em um arquivo XML. Nesta nova versão, é utilizado um banco de dados que armazena a localização dos links, defeitos e notas. Para facilitar o carregamento dos *mockups*, os novos arquivos permitem armazenar imagens. Desta forma, a ferramenta pode gerar os relatórios diretamente do arquivo de banco de dados que foi salvo pela ferramenta.

Na próxima seção é apresentado o design de uma nova versão da ferramenta *Mockup DUE*. Esta nova versão pretende facilitar o uso da ferramenta por inspetores com pouco conhecimento na técnica *Web DUE*.

7.4 Proposta da nova versão da Ferramenta *Mockup Design Usability Evaluation*

A ideia de uma ferramenta que guie os inspetores novatos pelo processo de inspeção da técnica *Web DUE* surgiu dos resultados qualitativos da avaliação cooperativa descrita no Capítulo 6. Durante a avaliação, dois inspetores afirmaram que o pouco conhecimento no processo de inspeção da técnica *Web DUE* podia dificultar o uso da ferramenta por inspetores novatos. Surgiu então a ideia de gerar uma nova versão da ferramenta *Mockup DUE* voltada para inspetores com pouca experiência na aplicação da técnica *Web DUE*. A proposta de tal ferramenta visa diminuir os custos de treinamento na técnica *Web DUE* [10].

A proposta da *Mockup Guiding Design Usability Evaluation (Mockup G-DUE)* pretende separar cada uma das etapas do processo de detecção de defeitos da técnica *Web DUE* e apresentá-las ao inspetor uma de cada vez. Para isto, foram propostas modificações na versão atual da ferramenta *Mockup DUE*. A seguir, são apresentados os *mockups* dessa nova proposta da ferramenta *Mockup DUE* para inspetores novatos.

Na Figura 7-8, é apresentada a versão atual da ferramenta *Mockup DUE*. Nessa figura, está marcada a área que será utilizada pela nova versão da ferramenta (*Mockup G-DUE*) para guiar os inspetores pelo processo de inspeção. Esta área, que inicialmente servia para apresentar as zonas e itens de verificação da técnica *Web DUE*, será substituída por um passo a passo do processo de inspeção da técnica.

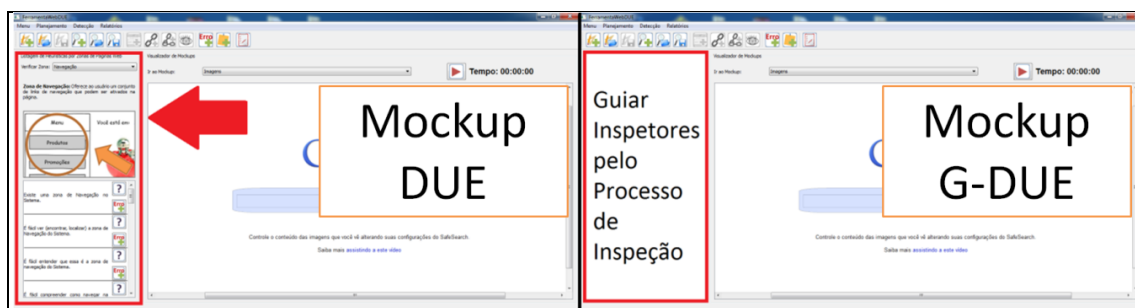


Figura 7-8 Localização das mudanças na proposta da *Mockup G-DUE*.

A Figura 7-9 apresenta os itens das primeiras etapas do processo de inspeção da técnica *Web DUE* sendo guiados pela proposta da ferramenta *Mockup G-DUE*.

Inicialmente os inspetores são questionados sobre as zonas de páginas *Web* presentes no *mockup* que está sendo avaliado (Figura 7-9 parte A). Dessa vez os inspetores precisam marcar quais são as zonas presentes no *mockup*. Em caso de dúvida, os inspetores também podem clicar no botão com o símbolo de interrogação para obter uma descrição mais detalhada sobre cada zona. Isto pode permite aos inspetores aprender quais são as zonas e a identificá-las. Quando o inspetor terminar de marcar todas as zonas de páginas *Web* presentes no *mockup* avaliado, a ferramenta apresentará os itens de verificação de cada zona, pedindo ao inspetor que verifique se esses são respeitados no *mockup*. Os exemplos/esclarecimentos também estão disponíveis nesta tela, permitindo ao inspetor tirar dúvidas e se familiarizar com os itens, como pode ser observado na Figura 7-9 parte B.

Seleção de Zonas (Parte A):

- ? Navegação
- ? Estado do Sistema
- ? Ajuda
- ? Informação
- ? Serviços
- ? Informação do Usuário
- ? Acesso Direto
- ? Entrada de Dados
- ? Instituição
- ? Customização

Verificação de Itens (Parte B):

Respeito à Zona de Informação, o mockup se adere às seguintes características?

- É possível reconhecer o significado dos termos (palavras ou símbolos) utilizados. ?
- As informações do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema. ?
- A interface do sistema permite ao usuário visualizar informações chave. ?
- Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real. ?

Callout Box: Os termos (palavras ou símbolos) utilizados pelo sistema são de fácil entendimento no contexto em que são usados.

Figura 7-9 Identificando zonas e verificando itens na proposta da *Mockup G-DUE*.

Na Figura 7-10, é apresentado o processo de identificação de problemas de usabilidade na proposta da ferramenta *Mockup G-DUE*. A técnica *Web DUE* requer que os inspetores indiquem se acham ou não que a violação de um item de verificação é um problema de usabilidade. Por esse motivo, a nova proposta da ferramenta solicitará aos inspetores que revejam cada um dos itens de verificação violado. Como pode ser observado na Parte A da Figura 7-10, a ferramenta pergunta a opinião do inspetor sobre

os itens que ele marcou como não conformes. Se o inspetor informar que o item violado é um problema, a ferramenta solicitará que seja descrito o porquê de tal resposta. Assim que o inspetor indicar que considerou um problema de usabilidade, a ferramenta gerará um defeito que deverá ser apontado no *mockup* (ver Figura 7-10 Parte B).

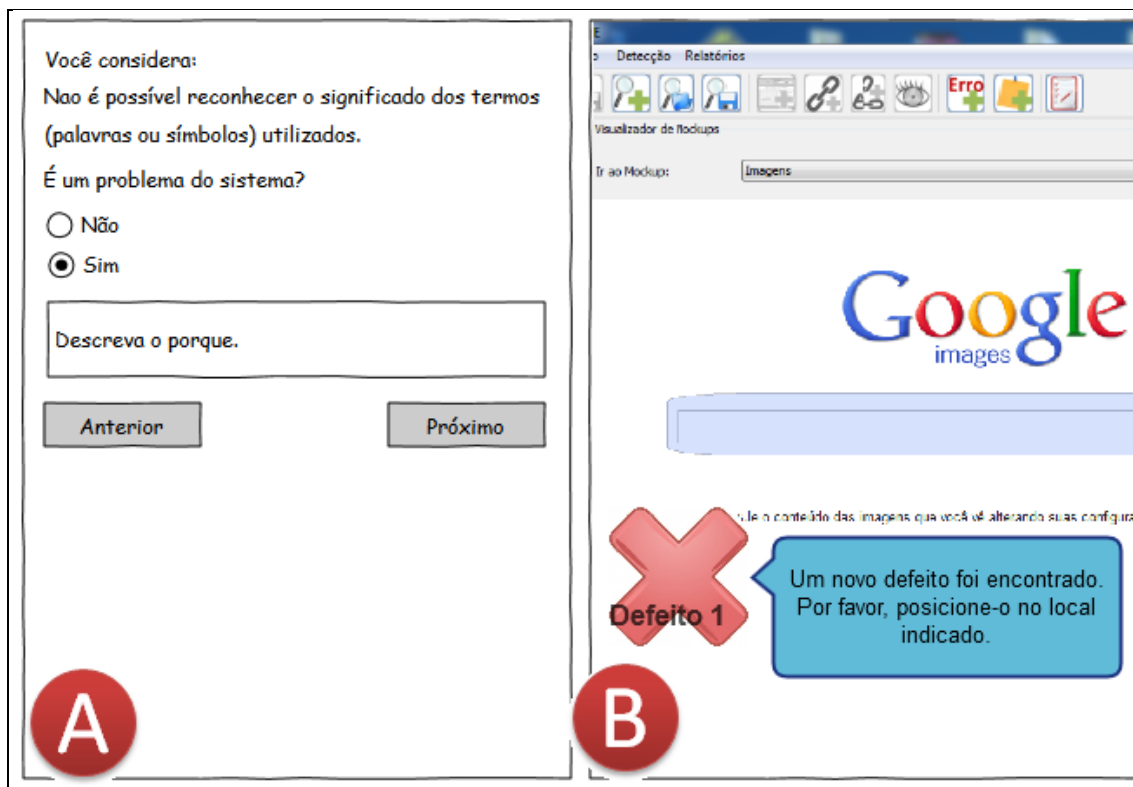


Figura 7-10 Identificando de problemas de usabilidade na *Mockup G-DUE*.

As outras funcionalidades da ferramenta trabalham da mesma forma descrita no Capítulo 5. Com estas mudanças espera-se que os inspetores novatos no uso da técnica *Web DUE* possam utilizá-la sem treinamento e a um baixo custo.

7.5 Considerações Finais

Este capítulo apresentou uma análise detalhada dos resultados obtidos nos estudos de viabilidade da técnica *Web DUE* e da ferramenta *Mockup DUE*. Esta análise serviu para identificar oportunidades de melhoria em ambas às propostas e assim melhorar o processo de inspeção da técnica *Web DUE* e a usabilidade da ferramenta *Mockup DUE* ao serem utilizadas durante inspeções de usabilidade.

Os resultados do estudo de viabilidade da técnica *Web DUE* apontaram que os inspetores tiveram dificuldade em entender e apontar defeitos com alguns dos itens de verificação da técnica. Esta dificuldade aconteceu por causa das instruções de uso da

técnica *Web DUE* e a falta de informações nos itens de verificação e exemplos/esclarecimentos da técnica. Como consequência, os inspetores apontaram defeitos de usabilidade que se tornaram falsos positivos, mesmo sabendo que não eram problemas de usabilidade reais. Além disso, alguns dos itens de verificação tinham descrições que fizeram os inspetores novatos achar defeitos que realmente não existiam.

Para melhorar o desempenho da técnica *Web DUE* foram feitas mudanças com base nesses resultados. Dessa forma, novas instruções foram adicionadas, e alguns itens de verificação e exemplos/esclarecimentos tiveram suas descrições alteradas. Apesar dos indícios obtidos no estudo de viabilidade da técnica, novos estudos devem ser executados para verificar se estas modificações foram suficientes e se não existem outras oportunidades de melhoria.

Com respeito à ferramenta *Mockup DUE*, as alterações sugeridas pelos inspetores durante o estudo de viabilidade foram implementadas. Além disso, os inspetores que participaram da avaliação cooperativa descrita no Capítulo 6 afirmaram que inspetores novatos na técnica *Web DUE* teriam dificuldade na utilização da ferramenta. Como consequência, foi proposta uma nova versão da ferramenta *Mockup DUE*, a *Mockup G-DUE* que tem por objetivo direcionar inspetores novatos pelo processo de inspeção da técnica *Web DUE*.

No próximo capítulo serão apresentadas considerações finais e contribuições deste trabalho, assim como as perspectivas de trabalhos futuros para continuação desta pesquisa.

8 Conclusões e Perspectivas Futuras

Neste capítulo as conclusões desta dissertação são apresentadas, resumindo sua motivação e proposta, apresentando as suas contribuições. As perspectivas futuras fornecem a direção para que seja dada continuidade ao trabalho relacionado à técnica Web DUE e ao apoio às inspeções de usabilidade aplicações Web durante as primeiras etapas do processo de desenvolvimento.

8.1 Epílogo

Esta dissertação de mestrado apresentou uma pesquisa sobre qualidade de aplicações Web. Nesta pesquisa foram adotadas as zonas próprias de páginas Web [13] para guiar o processo de inspeção de usabilidade de *mockups* Web e garantir sua qualidade em termos de usabilidade.

Para a proposta inicial, avaliação e evolução da técnica Web DUE – *Web Design Usability Evaluation*, esta pesquisa se baseou em uma adaptação da metodologia baseada em evidências [20][35]. A técnica Web DUE foi proposta com base nos resultados da extensão do mapeamento sistemático de Fernandes *et al.* [12]. Nesta extensão, conseguiu-se identificar três características que novos métodos de inspeção de usabilidade devem possuir para apoiar a indústria de desenvolvimento de software no que diz respeito a inspeções de usabilidade: (a) encontrar problemas de usabilidade nas primeiras etapas do processo de desenvolvimento; (b) guiar o processo de inspeção para encontrar problemas de aplicações Web e ajudar a corrigi-los; e (c) oferecer apoio ferramental para melhorar o desempenho da inspeção.

Como resultado da integração destas características surgiu a técnica Web DUE. Esta técnica foi desenvolvida de forma que pudesse ser aplicada em *mockups* de aplicações Web. A técnica Web DUE guia os inspetores pelo processo de inspeção através de zonas próprias de páginas Web, e é apoiada pela ferramenta *Mockup Design Usability Evaluation – Mockup DUE*, que simula a interação entre o usuário e o sistema para que o inspetor possa focar na identificação de problemas de usabilidade.

Até o presente momento, a técnica *Web DUE* e a ferramenta *Mockup DUE* foram avaliadas através de dois estudos de viabilidade. O primeiro estudo visou responder se a técnica *Web DUE* é viável em termos do número de defeitos identificados e o tempo gasto para encontrá-los. Os resultados ofereceram indícios da viabilidade da técnica quando comparada com a técnica base, a técnica *Web Design Perspectives - WDP* [6]. A técnica *Web DUE* atingiu uma média de eficácia de 22,32% de defeitos encontrados do total de defeitos conhecido, e uma média de eficiência de 10,89 defeitos por hora. Além disso, os resultados do estudo de viabilidade da ferramenta apontaram uma média de nível de satisfação de usuário de 7,7 pontos de um total de 10. Os resultados qualitativos de ambos os estudos também indicaram que tanto a técnica quanto a ferramenta foram consideradas apropriadas para a inspeção de *mockups* de aplicações *Web*.

8.2 Contribuições

As principais contribuições desta dissertação são:

- Estudo da área de Inspeções de Usabilidade de Aplicações *Web*, visando identificar o estado da arte em relação a inspeções de usabilidade dessas aplicações. Foi realizada uma extensão de um mapeamento sistemático para caracterizar os métodos de inspeção de usabilidade utilizados na avaliação de aplicações *Web*, o que implicou em:
 - Estabelecimento de uma base de conhecimento sobre métodos de inspeção de usabilidade, que foi utilizada nesta pesquisa.
 - Proposta de características a serem adotadas por novas técnicas de inspeção de usabilidade para apoiar a indústria de desenvolvimento de software no que diz respeito a inspeções de aplicações *Web*.
- Desenvolvimento de tecnologias de apoio à inspeção de usabilidade:
 - Definição de uma técnica de inspeção de usabilidade específica para *mockups* de aplicações *Web*, que pode ser empregada durante as primeiras etapas do processo de desenvolvimento.
 - Desenvolvimento de uma ferramenta de apoio ao processo de inspeção de *mockups* de aplicações *Web*, que utiliza a técnica *Web DUE* abstraindo o processo de mapeamento de *mockups*.

- Proposta de uma nova versão da ferramenta de apoio ao processo de inspeção de *mockups* de aplicações *Web*, que pode guiar inspetores com pouco conhecimento no uso da técnica *Web DUE* pelo processo de inspeção de *mockups*.
- Definição de estudos de viabilidade para avaliação das tecnologias propostas que permitiram:
 - Avaliação e evolução da técnica *Web DUE*.
 - Avaliação e evolução da ferramenta *Mockup DUE*.
- Disseminação dos resultados obtidos nesta pesquisa. Ao longo deste trabalho de mestrado, foram publicados vários artigos descrevendo os resultados dos estudos secundários e primários realizados nesta pesquisa. A seguir é apresentada a lista de artigos aceitos descrevendo seu foco:
 - O artigo “*Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension*” [29] foi publicado no XI *Experimental Software Engineering Latin American Workshop (ESELAW 2012)*. Nesse artigo, descreve-se a extensão do mapeamento sistemático apresentada no Capítulo 2. Os autores deste artigo foram convidados a estendê-lo para uma edição especial com os melhores artigos do CiBSE e ESELAW 2012 no *CLEI Electronic Journal*.
 - O artigo “*Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications*” [31] foi publicado na 24th *International Conference on Software Engineering & Knowledge Engineering (SEKE 2012)*. Nesse artigo, descreve-se a primeira versão da técnica *Web DUE*, apresentada no Capítulo 3, decorrente dos resultados da extensão do mapeamento sistemático.
 - O artigo “*Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications*” [30] foi publicado no XXVI Simpósio Brasileiro de Engenharia de Software (SBES 2012). Nesse artigo, descreve-se o estudo de viabilidade da técnica *Web DUE* e os seus resultados como apresentado no Capítulo 4. Este artigo obteve o prêmio de “1º Melhor Trabalho”. Além disso, os autores deste artigo foram convidados a estendê-lo para o *Journal of Software Engineering Research and Development (JSERD)*.

8.3 Perspectivas Futuras

A realização deste trabalho de pesquisa levou ao desenvolvimento de uma técnica de inspeção de usabilidade voltada para *mockups* de aplicações *Web*, além de uma ferramenta de apoio ao processo de inspeção que permite mapear os *mockups* para que os inspetores foquem o esforço na identificação de problemas de usabilidade. Estes resultados abrem novas perspectivas de pesquisa que podem ser exploradas em trabalhos futuros. Os principais trabalhos futuros são detalhados a seguir.

8.3.1 Desenvolvimento e Avaliação da Ferramenta *Mockup G-DUE*

Conforme apresentado no Capítulo 7, os resultados qualitativos do estudo de viabilidade da ferramenta *Mockup DUE* motivaram a proposta de uma nova versão da ferramenta, desta vez específica para inspetores novatos na técnica *Web DUE*. Esta nova versão denominada *Mockup Guiding Design Usability Evaluation (Mockup G-DUE)*, pretende guiar os inspetores novatos em cada etapa do processo de inspeção da técnica *Web DUE*.

Na ferramenta *Mockup G-DUE*, devem ser feitas mudanças na interface e no design de interação para que seja possível executar cada etapa do processo de detecção de defeitos da técnica *Web DUE* separadamente, e de acordo com as ações do inspetor. Para isso devem ser desenvolvidas as telas descritas na Seção 7.4 e repensadas funcionalidades específicas para inspetores novatos. Os itens de verificação devem ser negados durante a etapa de identificação de defeitos para facilitar sua compreensão como um problema de usabilidade.

A ferramenta *Mockup G-DUE* também deve ser avaliada experimentalmente para verificar sua viabilidade ao ser usada por inspetores novatos na técnica *Web DUE*. Neste estudo experimental, pode ser feita uma comparação dos resultados obtidos de ambas as ferramentas (*Mockup DUE* e *Mockup G-DUE*) visando verificar se a ferramenta *Mockup G-DUE* atinge resultados similares com os da ferramenta base ao ser usada por inspetores novatos. Além disso, a análise de resultados qualitativos pode ajudar a identificar oportunidades de melhoria nesta nova versão da ferramenta.

8.3.2 Avaliações Adicionais

Apesar de terem sido realizados dois estudos experimentais para avaliar a viabilidade da técnica *Web DUE* e da ferramenta *Mockup DUE*, diferentes avaliações adicionais

podem ser executadas. Em primeiro lugar, podem ser repetidos os estudos realizados neste trabalho, porém aumentando o número de participantes, de forma a incrementar a significância dos resultados obtidos e reduzir as ameaças de conclusão do estudo.

Também se faz necessário continuar realizar outros tipos de estudo como proposto por Shull *et al.* [35] e Mafra *et al.* [20]. Estes estudos podem ajudar a entender melhor o processo de aplicação da técnica em conjunto com a ferramenta, e apoiar a identificação de melhorias não sugeridas nos dois estudos de viabilidade. Além disso, novos estudos comparando a técnica *Web DUE* com outras técnicas específicas para aplicações *Web*, como a *Heuristic evaluation of paper-based Web pages* (S01), poderiam ser realizados. A divulgação de estudos experimentais, apontando situações específicas nas quais cada técnica poderia ser utilizada, pode contribuir para uma maior utilização por parte da indústria de desenvolvimento de software, em particular no desenvolvimento de aplicações *Web* de qualidade no que diz respeito à usabilidade.

8.3.3 Transferência das Tecnologias para a Indústria

É necessário que a técnica *Web DUE* e a ferramenta *Mockup DUE* sejam aplicadas em um contexto industrial. A utilização das tecnologias propostas em uma empresa de desenvolvimento de software pode proporcionar dados relevantes para a sua melhoria. Além disso, podem ser realizados mais estudos de caso com diferentes tipos de aplicações *Web*, aumentando não somente a aplicação desta proposta em diferentes contextos, mas o número de participantes e com isso a validade estatística dos próximos estudos.

Espera-se que os resultados obtidos neste trabalho sejam utilizados para a promoção e melhoria do estado atual das pesquisas sobre qualidade de aplicações *Web*. Com o uso das tecnologias propostas neste trabalho, espera-se também apoiar a avaliação da usabilidade de aplicações *Web* durante as primeiras etapas do processo de desenvolvimento, melhorando sua qualidade a um baixo custo.

Referências

- [1] Akinkuolie, B.: A Cross-Platform Mobile Learning System Using QT SDK Framework. Proc. V International Conference on Genetic and Evolutionary Computing (ICGEC), Japan, 2011, 163-167.
- [2] Allen, M., Currie, L., Patel, S., Cimino, J.: Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology. Journal of Biomedical Informatics, 2006.
- [3] Basili, V., Rombach, H.: The tame project: towards improvement-oriented software environments. IEEE Transactions on Software Engineering, Volume 14, Issue 6, 1988.
- [4] Bolchini, D., Garzotto, F.: Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+. Proc. of the International Workshop on Web Usability and Accessibility (IWWUA'07), 2007.
- [5] Charette, R.: Why software fails. *IEEE Spectrum*, Volume 42, Issue 9, 2005.
- [6] Conte, T., Massollar, J., Mendes, E., Travassos, G.: Web usability inspection technique based on design perspectives. IET Software, Volume 3, Issue 2, 2009.
- [7] Cutrell, E., Guan, Z.: What are you looking for? An eye-tracking study of information usage in web search. Proc. Conference on human factors in computing systems CHI'2007, USA, 2007, pp. 407-416.
- [8] Donahue, G., Weinschenk, S., Nowicki, J.: Usability is Good Business. Compuware Corporation Research Report, 1999.
- [9] Dyba, T., Kampenes, V., Sjoberg, D.: A systematic review of statistical power in software engineering experiments. Information and Software Technology, Volume 48, Issue 8, 2006.
- [10] Fernandes, P., Conte, T., Bonifacio, B.: Improving a Web Usability Inspection Technique through an Observational Study. Proc. 24th International Conference on Software Engineering & Knowledge Engineering, USA, 2012, pp. 588-593.
- [11] Fernandez, A., Abrahao, S., Insfran, E.: Towards to the validation of a usability evaluation method for model-driven Web development. Proc. IV International Symposium on Empirical Software Engineering and Measurement, USA, 2010, pp 54-57.

- [12] Fernandez, A., Insfran, E., Abrahao, S.: Usability evaluation methods for the Web: A systematic mapping study. *Information and Software Technology*, Volume 53, Issue 8, 2011.
- [13] Fons, J., Pelechano, V., Pastor, O., Valderas, P., Torres, V.: Applying the OOWS model-driven approach for developing Web applications: The internet movie database case study. In: Rossi, G., Schwabe, D., Olsina, L., Pastor, O.: *Web Engineering: Modeling and Implementing Web Applications*, Springer, 2008.
- [14] Garzotto, F., Paolini, P., Schwabe, D.: HDM— A Model-Based Approach to Hypermedia Application Design *ACM Trans. Information Systems*, Volume 11, Issue 1, 1993.
- [15] Insfran, E., Fernandez, A.: A Systematic Review of Usability Evaluation in Web Development. *Proc. Second International Workshop on Web Usability and Accessibility*, New Zealand, 2008, pp. 81-91.
- [16] International Organization for Standardization, ISO/IEC 9241-11: Ergonomic Requirements for Office work with Visual Display Terminals (VDTs) – Part 11: Guidance on Usability (1998)
- [17] Jöreskog, K.: A general method for analysis of covariance structure. *Biometrika*, Volume 57, Issue 2, 1970.
- [18] Kappel, G., Proll, B., Reich, S., Retschitzegger, W.: An Introduction to Web Engineering. In: Kappel, G., Proll, B., Reich, S., Retschitzegger, W.: *Web Engineering: The Discipline of Systematic Development of Web Applications*, John Wiley & Sons, 2006.
- [19] Kumar, V.: Social Coupons as a Marketing Strategy: A Multifaceted Perspective. *Journal of the Academy of Marketing Science*, Volume 40, Issue 1, 2012.
- [20] Mafra, S., Barcelos, R., Travassos, G.: Aplicando uma Metodologia Baseada em Evidência na Definição de Novas Tecnologias de Software. *Proceedings of the 20th Brazilian Symposium on Software Engineering (SBES 2006)*, 2006.
- [21] Matera, M., Rizzo, F., Carughi, G. T.: Web Usability: Principles and Evaluation Methods. In: Mendes, E., Mosley, N.: *Web Engineering*, Springer, 2006.
- [22] Mendes, E., Mosley, N., Counsell, S.: The Need for Web Engineering: An Introduction. *Web Engineering*. In: Mendes, E., Mosley, N.: *Web Engineering*, Springer, 2006.

- [23] Milewski, A.: Software engineers and HCI practitioners learning to work together: a preliminary look at expectations. Proc. XVII Conference on Software Engineering Education and Training, USA, 2004, pp. 44-49.
- [24] Monk, A., Wright, P., Haber, J., and Davenport, L.: Improving your human-computer interface: A practical technique. Prentice Hall International (UK) Ltd, 1993.
- [25] Murugesan, S. Web Application Development: Challenges and the Role of Web Engineering. In Rossi, G., Schwabe, D., Olsina, L., Pastor, O.: Web Engineering: Modeling and Implementing Web Applications, Springer, 2008.
- [26] Nielsen, J.: Heuristic evaluation. Jakob Nielsen, Mack, R. L. (eds), 1994.
- [27] Offutt, J.: Quality attributes of Web software applications. IEEE Software: Special Issue on Software Engineering of Internet Software, Volume 19, Issue 2, 2002.
- [28] Polson, P., Lewis, C., Rieman, J., Wharton, C.: Cognitive walkthroughs: a method for theory-based evaluation of user interfaces. International Journal of Man-Machine Studies, 1992.
- [29] Rivero, L., Conte, T.: Characterizing Usability Inspection Methods through the Analysis of a Systematic Mapping Study Extension. Proc. 11th Experimental Software Engineering Latin American Workshop, Argentina, 2012.
- [30] Rivero, L., Conte, T.: Using an Empirical Study to Evaluate the Feasibility of a New Usability Inspection Technique for Paper Based Prototypes of Web Applications, Proc. 26th Brazilian Symposium on Software Engineering, Brazil, 2012, pp. 81-90.
- [31] Rivero, L., Conte, T.: Using the Results from a Systematic Mapping Extension to Define a Usability Inspection Method for Web Applications. Proc. 24th International Conference on Software Engineering & Knowledge Engineering, USA, 2012, pp. 582-587.
- [32] Rocha, H., Baranauska, M.: Design e Avaliação de Interfaces Humano-Computador (Livro), Nied, 2003.
- [33] Schaik, P., Ling, J.: Using on-line surveys to measure three key constructs of the quality of human-computer interaction in Web sites: psychometric properties and implications. International Journal of Human-Computer Studies, Volume 5, Issue 5, 2003.
- [34] Shneiderman B.: Designing the user interface. 3rd ed. Reading, MA: Addison-Wesley, 1998.
- [35] Shull, F., Carver, J., Travassos, G.: An empirical methodology for introducing software processes. ACM SIGSOFT Software Engineering Notes, 2001.

- [36] SPSS (2010). SPSS version 19.0.0 for Windows. Chicago, SPSS Inc.
- [37] Tanaka, E., Rocha, E.: Evaluation of Web Accessibility Tools. Proc. 10th Brazilian Symposium on Human Factors in Computing Systems and the 5th Latin American Conference on Human-Computer Interaction, Brazil, 2011, pp. 272 – 279.
- [38] Travassos, G., Shull, F., Fredericks, M., Basili, V.: Detecting defects in object-oriented designs: using reading techniques to increase software quality. Proc. XIV ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications, USA, 1999, pp. 47-56.
- [39] Triacca, L., Inversini, A., Bolchini, D.: Evaluating Web usability with MiLE+. Proc. 7th IEEE International Symposium on Web Site Evolution, Hungary, 2005, pp. 22-29.
- [40] Wohlin, C. Runeson, P., Host, M., Ohlsson, M., Regnell, B., Wessl, A.: Experimentation in software engineering: an introduction (Book), Kluwer Academic Publishers, 2000.
- [41] Zhang, Z., Basili, V., Shneiderman, B.: Perspective-based Usability Inspection: An Empirical Validation of Efficacy. Empirical Software Engineering, 1999.

Apêndice A: Referências da Extensão do Mapeamento Sistemático

- [S01] Allen, M., Currie, L., Patel, S., Cimino, J.: Heuristic evaluation of paper-based Web pages: A simplified inspection usability methodology. *Journal of Biomedical Informatics*, 2006.
- [S02] Rios, D., Vazquez, I., Rey, E., Bonillo, V., Del Rio, B.: An HTML analyzer for the study of web usability. *Proc. Of the IEEE International Conference on Systems, Man and Cybernetics*, 2009.
- [S03] Ardito, C., Lanzilotti, R., Buono, P., Piccinno, A.: A tool to support usability inspection. *Proc. of the Working Conference on Advanced Visual Interfaces*, 2006.
- [S04] Atterer, R., Schmidt, A.: Adding Usability to Web Engineering Models and Tools. *Proc. of the 5th International Conference on Web Engineering*, 2005.
- [S05] Basu, A.: Context-driven assessment of commercial Web sites. *Proc. of the 36th Annual Hawaii International Conference on System Sciences*, 2003.
- [S06] Blackmon, M., Polson, P., Kitajima, M., Lewis, C.: Cognitive walkthrough for the web. *Proc. Of the SIGCHI conference on Human factors in computing systems*, 2002.
- [S07] Blackmon, M., Kitajima, M., Polson, P.: Repairing usability problems identified by the cognitive walkthrough for the web. *Proc. of the SIGCHI conference on Human factors in computing systems*, 2003.
- [S08] Blackmon, M., Kitajima, M., Polson, P.: Tool for accurately predicting website navigation problems, non-problems, problem severity, and effectiveness of repairs. *Proc. of the SIGCHI conference on Human factors in computing systems*, 2005.
- [S09] Bolchini, D., Garzotto, F.: Quality of Web Usability Evaluation Methods: An Empirical Study on MiLE+. *Proc. of the International Workshop on Web Usability and Accessibility*, 2007.
- [S10] Burton, C., Johnston, L.: Will World Wide Web user interfaces be usable? *Proc. Of the Computer Human Interaction Conference*, 1998.
- [S11] Chattratichart, J., Brodie, J.: Applying user testing data to UEM performance metrics. *Proc. of the Conference on Human factors in computing systems*, 2004.
- [S12] Conte, T., Massollar, J., Mendes, E., Travassos, G.: Web usability inspection technique based on design perspectives. *IET Software*, 2009.

- [S13] Costabile, M., Matera, M.: Guidelines for hypermedia usability inspection. IEEE Multimedia, 2001.
- [S14] Filgueiras, L., Martins, S., Tambascia, C., Duarte, R.: Recoverability Walkthrough: An Alternative to Evaluate Digital Inclusion Interfaces. Proc. of the Latin American Web Congress, 2009.
- [S15] Fraternali, P., Tisi, M.: Identifying Cultural Markers for Web Application Design Targeted to a Multi-Cultural Audience. Proc. of the 8th International Conference on Web Engineering, 2008.
- [S16] Habuchi, Y., Kitajima, M., Takeuchi, H.: Comparison of eye movements in searching for easy-to-find and hard-to-find information in a hierarchically organized information structure. Proc. of the symposium on Eye tracking research & applications, 2008.
- [S17] Kirmani, S.: Heuristic Evaluation Quality Score (HEQS): Defining Heuristic Expertise. Journal of Usability Studies, 2008.
- [S18] Molina, F., Toval, A.: Integrating usability requirements that can be evaluated in design time into Model Driven Engineering of Web Information Systems. Advances in Engineering Software, 2009.
- [S19] Moraga, M., Calero, C., Piattini, M.: Ontology driven definition of a usability model for second generation portals. Proc. of the 1st International Workshop on Methods, Architectures & Technologies for e-Service Engineering, 2006.
- [S20] Oztekin, A., Nikov, A., Zaim, S.: UWIS: An assessment methodology for usability of web-based information systems. Journal of Systems and Software, 2009.
- [S21] Paganelli, L., Paterno, F.: Automatic reconstruction of the underlying interaction design of web applications. Proc. of the 14th international conference on Software engineering and knowledge engineering, 2002.
- [S22] Paolini, P.: Hypermedia, the Web and Usability issues. Proc. of IEEE International Conference on Multimedia Computing and Systems, 1999.
- [S23] Signore, O.: A comprehensive model for Web sites quality. Proc. of the Seventh IEEE International Symposium on Web Site Evolution, 2005.
- [S24] Thompson, A., Kemp, E.: Web 2.0: extending the framework for heuristic evaluation. Proc. of the 10th International Conference NZ Chapter of the ACM's Special Interest Group on Human-Computer Interaction, 2009.
- [S25] Triacca, L., Inversini, A., Bolchini, D.: Evaluating Web usability with MiLE+. Proc. of the Seventh IEEE International Symposium on Web Site Evolution, 2005.

[S26] Vanderdonckt, J., Beirekdar, A., Noirhomme- Fraiture, M.: Automated Evaluation of Web Usability and Accessibility by Guideline Review. Proc. of the 4th International Conference on Web Engineering, 2004.

Apêndice B: A Técnica *Web DUE* v1

Instruções para Identificação de Problemas Usando a Técnica *Web DUE*

1. Selecione um dos *mockups* da aplicação *Web* que deseja avaliar.
2. Identifique as zonas próprias de páginas *Web* do *mockup* selecionado.
Para cada zona identificada:
 - 2.1. Avalie se a zona do *mockup* se adere aos itens de verificação listados no *checklist*.
Para cada item de verificação:
 - 2.1.1. Se o *mockup* não estiver aderente ao item de verificação, um novo problema de usabilidade foi encontrado.
 - 2.1.1.1. Preencha “Não Ok” ao lado do item de verificação avaliado no *checklist*.
 - 2.1.1.2. Descreva o problema e classifique-o segundo a escala de severidade: não é problema, cosmético, leve, grave e catastrófico.
 - 2.1.2. Se o *mockup* estiver aderente ao item de verificação, não houve nenhum problema de usabilidade.
 - 2.1.2.1. Preencha “Ok” ao lado do item de verificação avaliado no *checklist*.
3. Repita os passos 1 e 2 para cada *mockup* que deve ser inspecionado.

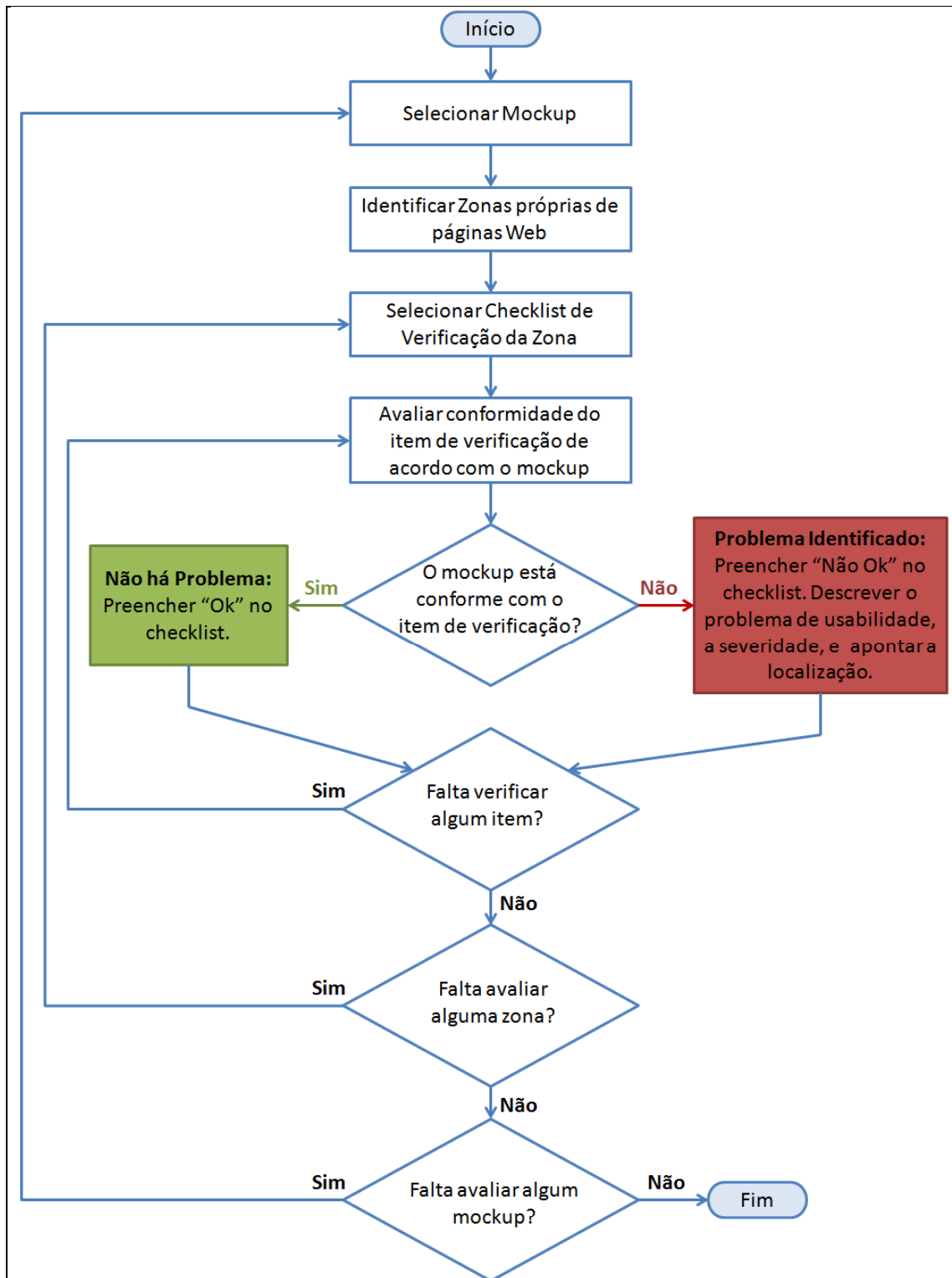
Importante:

- As zonas obrigatórias (Zona de Estado do Sistema, Zona de Navegação e Zona de Ajuda) sempre devem estar presentes nos *mockups* *Web*, e devem ser avaliadas.
- Se uma parte do *mockup* se repete ao longo da aplicação, então só é necessário avaliar as zonas relacionadas a esta parte do *mockup* uma vez. Caso sejam percebidas novas não conformidades relacionadas com a parte do *mockup* repetido, estas devem ser reportadas no *checklist* que foi utilizado para avaliar as zonas relacionadas a esta parte do *mockup*.
- A medida que novas partes dos *mockups* vão surgindo, novos *checklists* devem ser adicionados (embora sejam do mesmo tipo de zona). Por exemplo, se houverem dois

formulários diferentes cada um terá de ser avaliado em um *checklist* da zona de entrada de dados.

Fluxograma de Atividades do Processo de Detecção da Técnica *Web DUE*

v1.



Itens de Verificação da Zona de Navegação

Zona de Navegação		
Descrição		
Oferece ao usuário um conjunto de links de navegação que podem ser ativados na página.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Navegação no Sistema.	O sistema possui um menu de navegação.
A6, A7	É fácil ver (encontrar, localizar) a zona de Navegação do Sistema.	O menu de navegação é visível e fácil de encontrar a todo momento.
A6, A7	É fácil entender que essa é a zona de navegação do Sistema.	É fácil reconhecer que o menu de navegação permite o acesso a outras partes do sistema.
C6	É fácil compreender como navegar na aplicação.	O uso do menu é intuitivo: é fácil saber como interagir com o menu.
N7	É fácil navegar na zona de navegação.	A lógica da navegação é simples e as categorias fazem sentido.
C6	É fácil entender os termos (palavras ou símbolos) usados.	Por exemplo, no menu se utiliza a palavra "comprar" que é mais fácil de entender que a palavra "adquirir".
A2	As opções de navegação estão apresentadas em uma ordem natural e lógica.	Por exemplo, para ver o saldo em conta poupança. O sistema organiza o menu assim: Saldos - Investimentos - Poupança.
C2	Os termos (palavras ou símbolos) fazem sentido no domínio do problema e seguem as convenções do mundo real.	Por exemplo, dinheiro é representado com o símbolo "\$".
A5	É possível diferenciar entre os possíveis caminhos ao navegar pela aplicação.	Por exemplo, quando se percorre o caminho: cadastro - graduação - turmas, o usuário sabe que irá cadastrar turmas de graduação e não de pós-graduação.
N5	A interface previne a ocorrência de erros de navegação.	As opções de navegação disponíveis definem claramente que resultados ou estados serão atingidos.
A4	A navegação é padronizada ao longo da aplicação. Ou seja, os termos, cores e símbolos são os mesmos ao longo da aplicação.	Por exemplo, se ao selecionar uma opção no menu, ela é destacada, outras opções selecionadas também deverão ser destacadas ao longo da aplicação.

Zona de Navegação		
Descrição		
Oferece ao usuário um conjunto de links de navegação que podem ser ativados na página.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação é útil.	As informações são necessárias para a interação, não existem informações que atrapalham a navegação.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de errar o caminho:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao fluxo principal.
No caso de erro no sistema de navegação:		
A9	a) O sistema mostra uma mensagem de erro.	Se uma página não está disponível o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: A página não está disponível no momento", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: A página não está disponível no momento. Por favor, tente mais tarde ou entre em contato com o suporte fazendo clique aqui."

Itens de Verificação da Zona de Estado do Sistema

Zona de Estado do Sistema		
Descrição		
Oferece ao usuário informações sobre a sua localização na aplicação e qual foi o caminho seguido (sequencia de páginas navegadas) para chegar nesse estado.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
1	Existe uma zona de Estado do Sistema,	O sistema mostra um título relacionado à atividade que o usuário está realizando, a localização no site, ou o caminho seguido até o estado atual.
A1	O estado do sistema está sempre visível ao usuário.	A qualquer momento o usuário consegue ver qual é o estado do sistema.
A6, A7	É fácil entender que essa é a zona de estado do Sistema.	É fácil reconhecer a zona do estado do sistema.
C1, C2, C6	O estado do sistema está expresso em uma representação facilmente compreendida pelo usuário.	Os termos, símbolos, palavras ou frases fazem parte do domínio do problema e são facilmente entendidos.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Por exemplo, quando o usuário está fazendo uma compra, um título sugestivo da atividade (por exemplo: "Sua Compra") aparece na tela.
A2	O estado do sistema é apresentado de forma natural e lógica.	Por exemplo, um usuário vendo os horários de um filme pode ver: "Você está em: Rio de Janeiro - Estreias - Horários".
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para representar o estado do sistema.	Por exemplo, o estado comprar é representado pela palavra "comprar" e pelo símbolo carrinho de compras. Toda vez que se entre nesse estado o sistema não mudará o símbolo ou palavra.
A5	O estado do sistema permite ao usuário diferenciar entre diferentes estados.	O estado do sistema muda dependendo das atividades. Por exemplo: "Você está em Consulta de Produtos", "Você está em compra de produto".
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação é útil.	As informações são necessárias, não existem informações que atrapalham o

Zona de Estado do Sistema		
Descrição		
Oferece ao usuário informações sobre a sua localização na aplicação e qual foi o caminho seguido (sequencia de páginas navegadas) para chegar nesse estado.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		entendimento do estado do sistema
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Ajuda

Zona de Ajuda		
Descrição		
Oferece ao usuário informações sobre o uso da aplicação ou seu funcionamento.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Ajuda no Sistema.	Um botão onde possa ser acessada a ajuda do sistema. Esta ajuda deve permitir tirar dúvidas do sistema ou seu funcionamento.
A6, A10	É fácil ver (encontrar, localizar) a zona de Ajuda ou um botão/link para ter acesso a ela.	O botão/link de ajuda ou informações de ajuda são visíveis e fácil de encontrar a todo momento.
A6	É fácil entender que essa é a zona de ajuda ou um botão para ter acesso a ela.	É fácil reconhecer que o botão/link de ajuda leva a responder dúvidas sobre a aplicação, ou que a informação disponível na zona de ajuda serve para responder dúvidas sobre o sistema.
N10	É fácil acessar a ajuda e documentação.	É fácil obter acesso às informações que tirem dúvidas sobre como prosseguir nas tarefas de interesse do usuário.
Acesse a Ajuda do sistema		
A2, A5	As informações de ajuda estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.	São apresentadas perguntas/dúvidas frequentes e são organizadas segundo tópicos. Outras informações de ajuda relevantes também são agrupadas.
C2, C6, C10	A linguagem utilizada na ajuda está relacionada com o domínio do problema.	Os termos (palavras ou símbolos) utilizados pelo sistema são de fácil entendimento no contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados na ajuda seguem as convenções do mundo real.	Os termos (palavras ou símbolos) utilizados pelo sistema fazem sentido para os usuários.
A5	O usuário consegue identificar as diferentes dúvidas que podem ser tiradas no sistema.	Por exemplo, num sistema de compras, o usuário consegue diferenciar a ajuda relacionada a cadastro da ajuda relacionada a busca de produtos.
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para disponibilizar a ajuda no sistema.	Se a ajuda usa o termo "comprador" para se referir ao usuário, então o mesmo termo deverá ser usado para se referir ao usuário em qualquer outra

Zona de Ajuda		
Descrição		
Oferece ao usuário informações sobre o uso da aplicação ou seu funcionamento.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		parte da ajuda.
A7	A organização e disposição das informações de ajuda minimizam o tempo e esforço para responder dúvidas.	É localização da ajuda e a sua organização permitem tirar dúvidas mais rápido
A7, A8	Perguntas frequentemente respondidas estão acessíveis.	A ajuda possui uma seção de "FAQs - Frequently Asked Questions" para tirar dúvidas comuns à maioria dos usuários
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8, A10, c10	A informação é útil.	As informações ajudam ou direcionam para a execução de tarefas complexas ou o entendimento de conceitos não triviais. Não existem informações que atrapalham o processo de tirar dúvidas do usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Informação

Zona de Informação		
Descrição		
Oferece ao usuário informações armazenadas no banco de dados da aplicação. Estas informações podem ser texto, números, tabelas, imagens, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Informação no Sistema.	O sistema mostra informações necessárias para realizar uma atividade, ou exibe dados na forma de texto, relatórios, tabelas etc.
C6	É possível reconhecer o significado dos termos (palavras ou símbolos) utilizados.	Os termos (palavras ou símbolos) utilizados pelo sistema são de fácil entendimento no contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Os termos (palavras ou símbolos) fazem sentido e são fáceis de serem entendidas.
A2, A5	As informações do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.	Por exemplo, em um blog, as imagens relacionadas a cada tópico aparecem ordenadas e segundo o tópico ao que pertencem.
A6, A8	A interface do sistema permite ao usuário visualizar informações chave.	Informações relevantes devem ser destacadas ou facilmente visíveis. Num site de compras o preço do produto pode ter uma letra maior, ou letras em negrito.
A5	A interface facilita a distinção entre dados diferentes.	Ao disponibilizar dados, o sistema deve facilitar a identificação de contextos diferentes. Por exemplo, valor do produto em promoção, valor do produto sem desconto, datas em que pode ser pedido um desconto.
A7	A disposição da informação minimiza o esforço da sua leitura, entendimento.	Por exemplo, a informação de uma notícia pode ser mostrada assim: título, descrição, imagem, data e repórter; para facilitar sua leitura.
N7	A interface permite ao usuário diferentes formas de acesso as informações.	As informações podem ser visualizadas seguindo caminhos diferentes.
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para disponibilizar informações.	Ao usar um símbolo para representar alguma informação (por exemplo, moeda dólar = US\$), o mesmo deve ser

Zona de Informação		
Descrição		
Oferece ao usuário informações armazenadas no banco de dados da aplicação. Estas informações podem ser texto, números, tabelas, imagens, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		usado ao longo da aplicação.
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento de informações importantes para o usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Serviços

Zona de Serviços		
Descrição		
Oferece ao usuário acesso a operações que podem ser ativadas. A zona de serviços está geralmente inclusa dentro da zona de informação e as operações disponíveis estão relacionadas às informações mostradas na zona de informação.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Serviços no Sistema.	O sistema mostra as opções de funcionalidades que podem ser realizadas dentro da zona de informação. Botões/links que levam a funções como: cadastrar itens, buscar itens, verificar dados, etc.
A6	É fácil ver (encontrar, localizar) os serviços do sistema.	Os botões/links que ativam funcionalidades são fáceis de encontrar.
A6	É fácil entender que esses são os serviços do sistema.	É possível reconhecer que clicando em uma das opções de serviço, o sistema ativa uma funcionalidade.
C6	É fácil compreender como usar os serviços da aplicação.	Uma vez ativado um serviço é fácil entender o que ele faz e como usá-lo.
C6	É fácil entender as palavras/termos/símbolos usados na zona de serviços.	Por exemplo, o link "agenda" de uma aplicação de organização pessoal, sugere visualizar a funcionalidade de administrar compromissos na agenda pessoal. Os nomes devem ser fáceis de entender segundo o contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Por exemplo, o símbolo do botão "busca" é representado por uma lupa.
A2	Os serviços se apresentam de forma ordenada e lógica.	Por exemplo, dentro da área administrativa do sistema, o usuário pode ver as funcionalidades: cadastrar, editar e deletar clientes. É lógico para o usuário encontrar essas funcionalidades dentro da área administrativa do sistema.
A5	É possível diferenciar os diferentes serviços.	O sistema deve permitir diferenciar entre funcionalidades como: incluir produtos para uma compra, efetivar a

Zona de Serviços		
Descrição		
Oferece ao usuário acesso a operações que podem ser ativadas. A zona de serviços está geralmente inclusa dentro da zona de informação e as operações disponíveis estão relacionadas às informações mostradas na zona de informação.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		compra, salvar lista de presentes.
A4	A zona de Serviços é padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Ao usar um símbolo (como uma lupa para o serviço de busca) o mesmo deve ser usado ao longo da aplicação.
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de errar o serviço:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao fluxo principal.
No caso de erro no serviço selecionado:		
A9	a) O sistema mostra uma mensagem de erro.	Se um serviço não está disponível o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: O serviço só está disponível de 6:00 am até 8:00 pm", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Suponha que o horário seja 5:40 am, um bom exemplo seria: "Erro: O serviço só está disponível de 6:00 am até 8:00 pm. Por favor, tente daqui a 20 minutos".

Itens de Verificação da Zona de Informação do Usuário

Zona de Informação do Usuário		
Descrição		
Oferece ao usuário informações relacionadas ao usuário logado no sistema. Estas informações aparecem para usuários registrados no sistema.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Informação do Usuário.	O sistema mostra dados do usuário logado como: o login, nome, etc.
A6	É fácil ver (encontrar, localizar) a zona de Informação do Usuário.	É fácil enxergar e encontrar onde estão os dados do usuário logado.
A6	É fácil entender que os dados são do usuário logado no sistema.	Os dados disponibilizados indicam que usuário está usando o sistema no momento.
C6	É fácil entender se alguém está logado no sistema e quem ele é.	Por exemplo, caso não tenha ninguém logado o sistema mostra: "Entre". Se tiver alguém logado o sistema mostra: "Bem vindo: Fulano de Tal", ou "Você está logado como: fulano@email.com".
C6	É fácil entender as palavras/termos/símbolos usados para indicar quem está ou não logado no sistema.	Palavras como "Login", "Entrar" ou "Você está logado como:" são palavras que indicam se tem ou não um usuário logado. Estas palavras devem ser fáceis de entender.
C2	Para mostrar informação do usuário do sistema são usados termos e símbolos relacionados com o domínio da aplicação.	Por exemplo, Usuário: fulano@email.com.
A2	As informações do usuário são apresentadas de forma ordenada e lógica.	Por exemplo, num site de compras, os dados relevantes do usuário podem ser mostrados na ordem: E-mail, saldo disponível, pontos promocionais, etc.
A4	Os dados do usuário devem ser padronizados (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Ao usar um termo (como o e-mail para identificar o usuário) o mesmo deve ser usado ao longo da aplicação.
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento de informações

Zona de Informação do Usuário		
Descrição		
Oferece ao usuário informações relacionadas ao usuário logado no sistema. Estas informações aparecem para usuários registrados no sistema.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		importantes sobre o usuário logado no sistema.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Acesso Direto

Zona de Acesso Direto		
Descrição		
Contém funcionalidades Web comuns como: Login, Logout, Home, etc. Permite acesso direto a funcionalidades da aplicação que são mais usadas pelos usuários.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
7 - A7	Existe uma zona de Acesso Direto no Sistema.	Existem atalhos para acessar funcionalidades usadas pela maioria dos usuários ou de importância para o sistema. Por exemplo: Login, home, recados, etc.
A6	É fácil ver (encontrar, localizar) a zona de Acesso Direto no Sistema.	Os atalhos estão à vista para que os usuários possam vê-los
A6	É fácil entender que essa é a zona de Acesso Direto do Sistema.	É possível identificar os atalhos como tais, ou seja, é intuitivo que ao clicar em um o usuário é direcionado a uma parte funcional do sistema.
C6	É fácil entender as palavras/termos/símbolos usados para definir atalhos na zona de Acesso Direto.	Os atalhos usam termos que ajudam o usuário a identificar qual é a funcionalidade que será ativada.
C2	São usados termos e símbolos relacionados com o domínio da aplicação.	Por exemplo, o atalho para login é um botão de nome "Iniciar Seção". Os termos relacionados ao domínio da aplicação são de fácil entendimento para quem usa o sistema.
A2	Os atalhos se apresentam de forma ordenada e lógica.	Os atalhos não estão espalhados ou desorganizados.
A5	É possível diferenciar entre os atalhos.	Atalhos diferentes (Por exemplo, atalho de busca, atalho de login, etc.) possuem nomes/identificações diferentes.
A4	Os atalhos são padronizados (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Se um atalho é identificado com certo nome, símbolo ou cor, os mesmos devem ser mantidos para não confundir o usuário.
A7	A disposição dos atalhos aumenta a eficiência de uso, minimizando o esforço de encontrá-los.	O usuário não passará muito tempo procurando ou identificando os atalhos.
N7	A interface permite ao usuário diferentes formas de acesso às tarefas	As tarefas de interesse do usuário devem ser apoiadas pelo sistema, disponibilizando meios de acesso

Zona de Acesso Direto		
Descrição		
Contém funcionalidades Web comuns como: Login, Logout, Home, etc. Permite acesso direto a funcionalidades da aplicação que são mais usadas pelos usuários.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	principais.	rápido e/ou caminhos alternativos.
A8	Os atalhos são legíveis.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	Os atalhos são úteis.	As informações são necessárias, não existem informações que atrapalham o uso dos atalhos.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de escolher um atalho errado:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao estado anterior.
No caso de erro ao usar um atalho:		
A9	a) O sistema mostra uma mensagem de erro.	Se um atalho não funciona o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: Seu pedido não pode ser atendido no momento", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: Seu pedido não pode ser atendido no momento. Por favor, tente mais tarde ou entre em contato com o suporte fazendo clique aqui."

Itens de Verificação da Zona de Entrada de Dados

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Entrada de Dados no Sistema.	O sistema apresenta meios de inserção de dados: formulários, opções para selecionar itens, opções para marcar itens, ou meios de fornecer dados ao sistema a fim de realizar uma tarefa.
A6	É fácil ver (encontrar, localizar) onde entrar dados no sistema.	O usuário enxerga facilmente onde tem de informar dados.
A6	É fácil entender que essa zona é para entrar dados do Sistema.	Campos de preenchimento, seleção, etc. devem ser reconhecíveis, o usuário deve saber que eles servem para inserir dados.
C6	É fácil compreender como ingressar dados no sistema.	O meio utilizado para informar dados ao sistema deve ser intuitivo. Para preencher o nome, o formulário pode usar uma caixa de texto. Por exemplo para informar o sexo, a interface pode usar radio buttons "M" e "F".
C6	Avalie se é possível reconhecer o significado dos termos (palavras ou símbolos) utilizados.	O usuário deve conseguir identificar que dados estão sendo solicitados sem dificuldade. Por exemplo, um campo data, não devia possuir nomes ambíguos como "época", "tempo", etc.
A2	Os dados a serem informados pelo usuário (seja qual for o meio em que são fornecidos) estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.	Ao realizar uma compra o usuário deve informar por exemplo: item a ser comprado, quantidade e forma de pagamento. A ordem de preenchimento dos dados deve ser lógica e segundo a funcionalidade sendo realizada.
C2	É usada a linguagem do domínio do problema.	Os termos usados devem ser facilmente entendíveis e relacionados com as funcionalidades a serem executadas no sistema.

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
C2	Os termos (palavras ou símbolos) utilizados seguem as convenções do mundo real.	Os termos usados devem fazer sentido para os usuários e não devem ser usados termos que sejam incomuns aos usuários. Uma compra não deve ser chamada de "investimento", pois nomes pouco usuais dificultam o entendimento do usuário.
A4	Há aderência a convenções de plataforma e padrões de interface adotados em relação a layout, formatação e controles.	Se uma data é preenchida no formato: dia/mês/ano, esse formato de preenchimento deve ser mantido e não alterado ao longo da aplicação.
A4	Tarefas similares possuem formulários de preenchimento similares. O estilo de preenchimento deve ser padronizado.	Ao informar dados em tarefas similares, os formulários não podem mudar radicalmente. Isso permite ao usuário saber como preencher dados em determinadas tarefas.
A5	Os dados obrigatórios na entrada de dados estão claramente definidos.	Por exemplo o sistema usa "*" ou "(obrigatório)" ao lado dos campos obrigatórios.
A7	Avalie se a interface permite a utilização de mecanismos de busca de informação que auxiliem a entrada de dados obrigatória.	Por exemplo, na hora de inserir um endereço, o sistema pode permitir procurar endereços no google maps.
A5	A interface indica o formato correto para uma entrada de dados específica.	Os campos devem conter/fornecer dicas de preenchimento. Ex. Data: dd/mm/aa, Telefone: 9999-9999.
A6-N10	A interface do sistema permite ao usuário visualizar informações sobre o formato correto de entrada de dados.	As dicas de preenchimento devem ser facilmente visualizadas pelo usuário.
C10	É possível compreender as informações sobre o formato correto de entrada de dados.	As dicas de preenchimento devem ser facilmente entendíveis.
C6	Dicas sobre os termos (palavras ou símbolos) específicos do domínio do problema estão acessível aos usuários.	O dado "Cod. de Seg" do cartão de crédito pode possuir a seguinte dica: "Código de três dígitos localizado no verso do cartão de crédito." Esta dica deve ser facilmente visualizada.

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A6	É fácil reconhecer/visualizar dados já fornecidos.	Quando o usuário tiver informado um dado, o mesmo deve ficar disponível para visualização.
A5	Os dados a serem fornecidos são apresentadas de forma balanceada.	Os dados a serem fornecidos são solicitados em proporções adequadas. Não é necessário informar muitos dados de uma vez só.
A5	A interface facilita a distinção entre dados diferentes.	Em caso de serem solicitadas dois dados similares, cada um deve ser identificado. Por exemplo: Data de vencimento, data de pagamento.
A7	A disposição dos dados solicitados na interface do sistema aumenta a eficiência de uso.	Os dados são solicitados de forma ordenada e agrupados a fim de facilitar seu fornecimento.
N7	A interface permite ao usuário navegar com facilidade ao preencher os dados.	É fácil navegar dentro do mesmo formulário. Se o formulário é preenchido ao longo de várias páginas, deve ser fácil acessar as diferentes partes do formulário.
N7	A interface permite ao usuário a utilização de aceleradores ou atalhos na inserção de dados.	Por exemplo, um usuário pode salvar os dados do seu cartão de crédito para não preencher os mesmos dados a cada compra.
A8	Os campos são legíveis.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	Os campos são úteis.	Os dados a serem informados são necessários, não existem informações que atrapalham o preenchimento dos dados. Dados desnecessários não são marcados como obrigatórios.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de errar o preenchimento/seleção de um dado:		

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de preencher o dado novamente.
	No caso de erro ao informar um dado:	
A9	a) O sistema mostra uma mensagem de erro.	Se o usuário não preencheu um dado obrigatório ou preencheu errado o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: Data de nascimento é um campo obrigatório", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: Data de nascimento é um campo obrigatório, por favor verifique se preencheu o dado e de forma correta"
A4	e) Avalie se as mensagens de erros são apresentadas de forma consistente com os padrões de apresentação utilizados.	Os erros similares devem ser notificados no mesmo padrão para o usuário identificar o tipo de erro cometido.

Itens de Verificação da Zona de Instituição

Zona de Instituição		
Descrição		
Oferece ao usuário informações sobre a Instituição (companhia, entidade, organização, etc.) que é responsável pela aplicação Web. Geralmente mostra informações sobre o nome da companhia, logo, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona que mostra a Instituição.	A página mostra o nome, logo, etc., da instituição responsável.
A6	É fácil ver (encontrar, localizar) a zona que mostra dados da Instituição.	O usuário consegue enxergar o nome/logo da instituição.
A6	É fácil entender que essa é a zona que mostra dados da Instituição.	O usuário deve reconhecer que as imagens/palavras mostradas são referentes à instituição.
C6	É fácil entender qual é a instituição do sistema.	O logo/nome deve ser sugestivo e o usuário deve entender qual é o propósito do site.
C6	É fácil entender as palavras/termos/símbolos usados para indicar qual é a instituição do sistema.	É fácil entender a mensagem passada no logo, nome, etc. e o que ele representa.
A4	A instituição é apresentada de forma padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Se um logo é usado em uma página o mesmo deve ser mantido nas outras. Manter consistência e padrões.
A8	A informação é legível.	A letra/imagem usada é fácil de ler/ver com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento do usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Customização

Zona de Customização		
Descrição		
Contém informações sobre outros conteúdos que não podem ser classificados dentro das zonas anteriormente descritas. Geralmente é usada para mostrar conteúdo que não está relacionado ao domínio da aplicação como: propagandas, outras aplicações Web, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Customização no Sistema (Ex. Conteúdo não relacionado com o domínio da aplicação, aplicações externas, propaganda, etc.).	
A6	É fácil ver (encontrar, localizar) a zona de Customização.	
A6	É fácil entender que essa é a zona de Customização.	
C6	É fácil compreender o que pode ser feito na zona de Customização.	
C6	É fácil entender as palavras/termos/símbolos usados na zona de Customização.	
C2	Os termos e símbolos relacionados com o objetivo da zona de Customização (Ex. Uma propaganda apresentada faz sentido).	
A2	Os componentes da zona de Customização se apresentam de forma lógica e ordenada.	
A5	É possível diferenciar entre os possíveis componentes e serviços da zona de Customização.	
7	É possível tomar atalhos na área Customização para acelerar a interação	
A4	A zona de Customização é padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	
	O design da zona de Customização:	
8	a) É agradável (Gosto dele).	
A8	b) Não produz desconforto físico. (Ex. o uso de cores fortes, ou letras	

Zona de Customização		
Descrição		
Contém informações sobre outros conteúdos que não podem ser classificados dentro das zonas anteriormente descritas. Geralmente é usada para mostrar conteúdo que não está relacionado ao domínio da aplicação como: propagandas, outras aplicações Web, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	pequenas e ilegíveis).	
A8	c) É simples (Não existem informações que atrapalham o entendimento).	
	Caso ocorra um erro (Ex. tomar um caminho errado):	
9	a) O usuário conseguirá identificar que errou.	
9	b) O usuário conseguirá dizer que tipo de erro foi. (Ex. Click errado, escolher menu errado, etc.)	
3	c) O usuário poderá desfazê-lo.	
9	d) O usuário poderá corrigi-lo.	

Apêndice C: A Técnica *Web DUE* v2

Instruções para Identificação de Problemas Usando a Técnica *Web DUE*

1. Selecione um dos *mockups* da aplicação *Web* que deseja avaliar.
2. Identifique as zonas próprias de páginas *Web* do *mockup* selecionado.

Para cada zona identificada:

- 2.1. Avalie se a zona do *mockup* se adere aos itens de verificação listados no *checklist*.

Para cada item de verificação:

- 2.1.1. Avalie se o item de verificação se aplica ao *mockup* que está sendo inspecionado. Isto é, as pré-condições necessárias para avaliar o item estão sendo respeitadas pelo *mockup*.

- 2.1.1.1. Se o *Mockup* **atender as pré-condições, avalie se a não conformidade pode ocasionar um problema** de usabilidade na aplicação *Web*.

- 2.1.1.1.1. Se você considerar que a não conformidade é **um problema**:

- 2.1.1.1.1.1. Preencha “Não Ok” ao lado do item de verificação avaliado no *checklist*.

- 2.1.1.1.1.2. Descreva o problema e classifique-o segundo a escala de severidade: não é problema, cosmético, leve, grave e catastrófico.

- 2.1.1.1.2. Se você considerar que a não conformidade **não é um problema**:

- 2.1.1.1.2.1. Preencha “Ok” ao lado do item de verificação avaliado no *checklist*.

- 2.1.1.2. Se o *Mockup* **não atender as pré-condições, não houve nenhum problema** de usabilidade.

- 2.1.1.2.1. Preencha “Ok” ao lado do item de verificação avaliado no *checklist*.

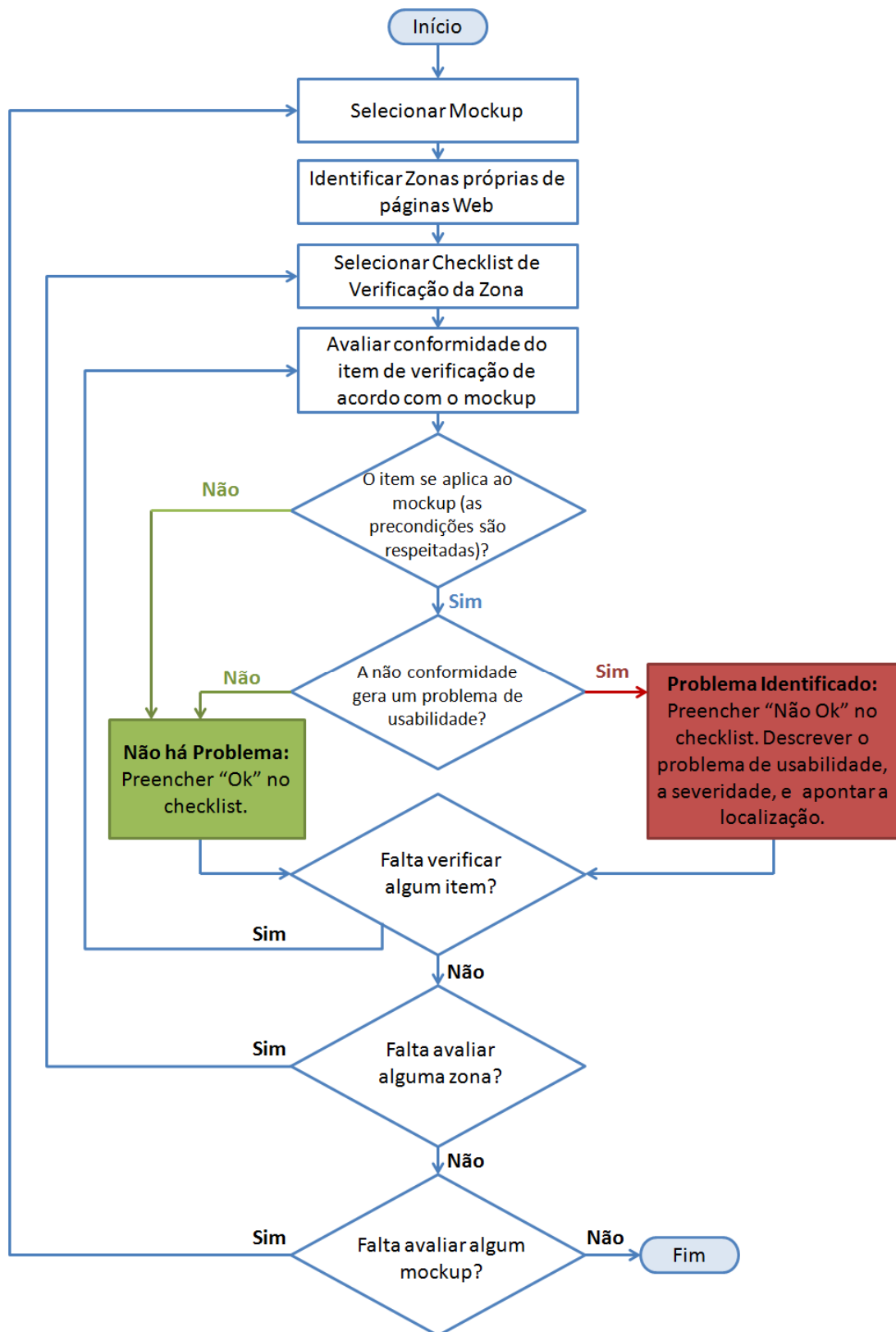
3. Repita os passos 1 e 2 para cada *mockup* que deve ser inspecionado.

Importante:

- As zonas obrigatórias (Zona de Estado do Sistema, Zona de Navegação e Zona de Ajuda) sempre devem estar presentes nos *mockups* Web, e devem ser avaliadas.
- Se uma parte do *mockup* se repete ao longo da aplicação, então só é necessário avaliar as zonas relacionadas a esta parte do *mockup* uma vez. Caso sejam percebidas novas não conformidades relacionadas com a parte do *mockup* repetido, estas devem ser reportadas no *checklist* que foi utilizado para avaliar as zonas relacionadas a esta parte do *mockup*.
- A medida que novas partes dos *mockups* vão surgindo, novos *checklists* devem ser adicionados (embora sejam do mesmo tipo de zona). Por exemplo, se houverem dois formulários diferentes cada um terá de ser avaliado em um *checklist* da zona de entrada de dados.

Fluxograma de Atividades do Processo de Detecção da Técnica Web DUE

v2.



Itens de Verificação da Zona de Navegação

Zona de Navegação		
Descrição		
Oferece ao usuário um conjunto de links de navegação que podem ser ativados na página.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Navegação no Sistema.	O sistema possui um conjunto de links de navegação que podem ser usados para se deslocar de uma página a outra.
A6, A7	É fácil ver (encontrar, localizar) a zona de Navegação do Sistema.	O menu de navegação é visível e fácil de encontrar a todo momento.
A6, A7	É fácil entender que essa é a zona de navegação do Sistema.	É fácil reconhecer que as opções de navegação disponibilizadas no sistema (links, menus, botões) permitem o acesso a outras partes do sistema.
C6	É fácil compreender como navegar na aplicação.	O uso do menu é intuitivo: é fácil saber como interagir com o menu.
N7	É fácil navegar na zona de navegação.	A lógica da navegação é simples e as categorias fazem sentido.
C6	É fácil entender os termos (palavras ou símbolos) usados.	Por exemplo, no menu se utiliza a palavra "comprar" que é mais fácil de entender que a palavra "adquirir".
A2	Se houver opções de navegação: As opções estão apresentadas em uma ordem natural e lógica.	Por exemplo, para ver o saldo em conta poupança. O sistema organiza o menu assim: Saldos - Investimentos - Poupança.
C2	Os termos (palavras ou símbolos) fazem sentido no domínio do problema e seguem as convenções do mundo real.	Por exemplo, dinheiro é representado com o símbolo "\$".
A5	Se for possível escolher diferentes caminhos de navegação: É possível diferenciá-los ao navegar.	Por exemplo, quando se percorre o caminho: cadastro - graduação - turmas, o usuário sabe que irá cadastrar turmas de graduação e não de pós-graduação.
N5	A interface previne a ocorrência de erros de navegação.	As opções de navegação disponíveis definem claramente que resultados ou estados serão atingidos.
A4	A navegação é padronizada ao longo da aplicação. Ou seja, os termos, cores e símbolos são os mesmos ao longo da	Por exemplo, se ao selecionar uma opção no menu, ela é destacada, outras opções selecionadas também deverão

Zona de Navegação		
Descrição		
Oferece ao usuário um conjunto de links de navegação que podem ser ativados na página.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	aplicação.	ser destacadas ao longo da aplicação.
A8	A informação disponibilizada para navegar é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação disponibilizada para navegar é útil.	As informações são necessárias para a interação, não existem informações que atrapalham a navegação.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de errar o caminho:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao fluxo principal.
No caso de erro ao navegar pelo sistema:		
A9	a) O sistema mostra uma mensagem de erro.	Se uma página não está disponível o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: A página não está disponível no momento", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: A página não está disponível no momento. Por favor, tente mais tarde ou entre em contato com o suporte fazendo clique aqui."

Itens de Verificação da Zona de Estado do Sistema

Zona de Estado do Sistema		
Descrição		
Oferece ao usuário informações sobre a sua localização na aplicação e qual foi o caminho seguido (sequencia de páginas navegadas) para chegar nesse estado.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
1	Existe uma zona de Estado do Sistema,	O sistema mostra um título relacionado à atividade que o usuário está realizando, a localização no site, ou o caminho seguido até o estado atual.
A1	O estado do sistema está sempre visível ao usuário.	A qualquer momento o usuário consegue ver qual é o estado do sistema.
A6, A7	É fácil entender que essa é a zona de estado do Sistema.	É fácil reconhecer a zona do estado do sistema.
C1, C2, C6	O estado do sistema está expresso em uma representação facilmente compreendida pelo usuário.	Os termos, símbolos, palavras ou frases fazem parte do domínio do problema e são facilmente entendidos.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Por exemplo, quando o usuário está fazendo uma compra, um título sugestivo da atividade (por exemplo: "Sua Compra") aparece na tela.
A2	O estado do sistema é apresentado de forma natural e lógica.	Por exemplo, um usuário vendo os horários de um filme pode ver: "Você está em: Rio de Janeiro - Estreias - Horários".
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para representar o estado do sistema.	Por exemplo, o estado comprar é representado pela palavra "comprar" e pelo símbolo carrinho de compras. Toda vez que se entre nesse estado o sistema não mudará o símbolo ou palavra.
A5	Se houver estados diferentes: O sistema permite diferenciá-los.	O estado do sistema muda dependendo das atividades. Por exemplo: "Você está em Consulta de Produtos", "Você está em compra de produto".
A8	A informação disponibilizada sobre o estado do sistema é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação disponibilizada sobre o estado do sistema é útil.	As informações são necessárias, não existem informações que atrapalham o

Zona de Estado do Sistema		
Descrição		
Oferece ao usuário informações sobre a sua localização na aplicação e qual foi o caminho seguido (sequencia de páginas navegadas) para chegar nesse estado.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		entendimento do estado do sistema
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Ajuda

Zona de Ajuda		
Descrição		
Oferece ao usuário informações sobre o uso da aplicação ou seu funcionamento.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Ajuda no Sistema.	Existe a opção de tirar dúvidas relacionadas ao sistema e seu funcionamento. Esta ajuda pode ser acessada clicando em botões, ou aparecer quando o usuário passar o mouse nos elementos da interface.
A6, A10	É fácil ver (encontrar, localizar) a zona de Ajuda ou um botão/link para ter acesso a ela.	O botão/link de ajuda ou informações de ajuda são visíveis e fácil de encontrar a todo momento.
A6	É fácil entender que essa é a zona de ajuda ou um botão para ter acesso a ela.	É fácil reconhecer que o botão/link de ajuda leva a responder dúvidas sobre a aplicação, ou que a informação disponível na zona de ajuda serve para responder dúvidas sobre o sistema.
N10	É fácil acessar a ajuda e documentação.	É fácil obter acesso às informações que tirem dúvidas sobre como prosseguir nas tarefas de interesse do usuário.
Se for possível, acesse a Ajuda do sistema.		
A2, A5	Se houver informações de ajuda: as informações de ajuda estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.	São apresentadas perguntas/dúvidas frequentes e são organizadas segundo tópicos. Outras informações de ajuda relevantes também são agrupadas.
C2, C6, C10	A linguagem utilizada na ajuda está relacionada com o domínio do problema.	Os termos (palavras ou símbolos) utilizados pelo sistema são de fácil entendimento no contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados na ajuda seguem as convenções do mundo real.	Os termos (palavras ou símbolos) utilizados pelo sistema fazem sentido para os usuários.
A5	O usuário consegue identificar as diferentes dúvidas que podem ser tiradas no sistema.	Por exemplo, num sistema de compras, o usuário consegue diferenciar a ajuda relacionada a cadastro da ajuda relacionada a busca de produtos.

Zona de Ajuda		
Descrição		
Oferece ao usuário informações sobre o uso da aplicação ou seu funcionamento.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para disponibilizar a ajuda no sistema.	Se a ajuda usa o termo "comprador" para se referir ao usuário, então o mesmo termo deverá ser usado para se referir ao usuário em qualquer outra parte da ajuda.
A7	A organização e disposição das informações de ajuda minimizam o tempo e esforço para responder dúvidas.	É localização da ajuda e a sua organização permitem tirar dúvidas mais rápido
A7, A8	Perguntas frequentemente respondidas estão acessíveis.	A ajuda possui uma seção de "FAQs - Frequently Asked Questions" para tirar dúvidas comuns à maioria dos usuários
A8	A informação para ajudar o usuário é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8, A10, c10	A informação para ajudar o usuário é útil.	As informações ajudam ou direcionam para a execução de tarefas complexas ou o entendimento de conceitos não triviais. Não existem informações que atrapalham o processo de tirar dúvidas do usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Informação

Zona de Informação		
Descrição		
Oferece ao usuário informações armazenadas no banco de dados da aplicação. Estas informações podem ser texto, números, tabelas, imagens, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Informação no Sistema.	O sistema mostra informações necessárias para realizar uma atividade, ou exibe dados na forma de texto, relatórios, tabelas etc.
C6	É possível reconhecer o significado dos termos (palavras ou símbolos) utilizados.	Os termos (palavras ou símbolos) utilizados pelo sistema são de fácil entendimento no contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Os termos (palavras ou símbolos) fazem sentido e são fáceis de serem entendidas.
A2, A5	As informações do sistema estão apresentadas em uma ordem natural e lógica segundo os conceitos do domínio do problema.	Por exemplo, em um blog, as imagens relacionadas a cada tópico aparecem ordenadas e segundo o tópico ao que pertencem.
A6, A8	A interface do sistema permite ao usuário visualizar informações chave.	Informações relevantes devem ser destacadas ou facilmente visíveis. Num site de compras o preço do produto pode ter uma letra maior, ou letras em negrito.
A5	A interface facilita a distinção entre dados diferentes.	Ao disponibilizar dados, o sistema deve facilitar a identificação de contextos diferentes. Por exemplo, valor do produto em promoção, valor do produto sem desconto, datas em que pode ser pedido um desconto.
A7	A disposição da informação minimiza o esforço da sua leitura, entendimento.	Por exemplo, a informação de uma notícia pode ser mostrada assim: título, descrição, imagem, data e repórter; para facilitar sua leitura.
N7	A interface permite ao usuário diferentes formas de acesso as informações.	As informações podem ser visualizadas seguindo caminhos diferentes.
A4	Existe padronização. A mesma terminologia, símbolos e formato são usados para disponibilizar informações.	Ao usar um símbolo para representar alguma informação (por exemplo, moeda dólar = US\$), o mesmo deve ser

Zona de Informação		
Descrição		
Oferece ao usuário informações armazenadas no banco de dados da aplicação. Estas informações podem ser texto, números, tabelas, imagens, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
		usado ao longo da aplicação.
A8	A informação disponibilizada no sistema é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação disponibilizada no sistema é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento de informações importantes para o usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Serviços

Zona de Serviços		
Descrição		
Oferece ao usuário acesso a operações que podem ser ativadas. A zona de serviços está geralmente inclusa dentro da zona de informação e as operações disponíveis estão relacionadas às informações mostradas na zona de informação.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Serviços no Sistema.	O sistema mostra as opções de funcionalidades que podem ser realizadas dentro da zona de informação. Botões/links que levam a funções como: cadastrar itens, buscar itens, verificar dados, etc. Opções relacionadas com as informações do sistema.
A6	É fácil ver (encontrar, localizar) os serviços do sistema.	Os botões/links que ativam funcionalidades são fáceis de encontrar.
A6	É fácil entender que esses são os serviços do sistema.	É possível reconhecer que clicando em uma das opções de serviço, o sistema ativa uma funcionalidade.
C6	É fácil compreender como usar os serviços da aplicação.	Uma vez ativado um serviço é fácil entender o que ele faz e como usá-lo.
C6	É fácil entender as palavras/termos/símbolos usados na zona de serviços.	Por exemplo, o link "agenda" de uma aplicação de organização pessoal, sugere visualizar a funcionalidade de administrar compromissos na agenda pessoal. Os nomes devem ser fáceis de entender segundo o contexto em que são usados.
C2	Os termos (palavras ou símbolos) utilizados pelo sistema seguem as convenções do mundo real.	Por exemplo, o símbolo do botão "busca" é representado por uma lupa.
A2	Os serviços se apresentam de forma ordenada e lógica.	Por exemplo, dentro da área administrativa do sistema, o usuário pode ver as funcionalidades: cadastrar, editar e deletar clientes. É lógico para o usuário encontrar essas funcionalidades dentro da área administrativa do sistema.

Zona de Serviços		
Descrição		
Oferece ao usuário acesso a operações que podem ser ativadas. A zona de serviços está geralmente inclusa dentro da zona de informação e as operações disponíveis estão relacionadas às informações mostradas na zona de informação.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A5	Se houver diferentes serviços: É possível diferenciá-los.	O sistema deve permitir diferenciar entre funcionalidades como: incluir produtos para uma compra, efetivar a compra, salvar lista de presentes.
A4	A zona de Serviços é padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Ao usar um símbolo (como uma lupa para o serviço de busca) o mesmo deve ser usado ao longo da aplicação.
A8	A informação sobre os serviços é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de selecionar o serviço errado:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao fluxo principal.
No caso de erro no serviço selecionado:		
A9	a) O sistema mostra uma mensagem de erro.	Se um serviço não está disponível o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: O serviço só está disponível de 6:00 am até 8:00 pm", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Suponha que o horário seja 5:40 am, um bom exemplo seria: "Erro: O serviço só está disponível de 6:00 am até 8:00 pm. Por favor, tente daqui a 20 minutos".

Itens de Verificação da Zona de Informação do Usuário

Zona de Informação do Usuário		
Descrição		
Oferece ao usuário informações relacionadas ao usuário logado no sistema. Estas informações aparecem para usuários registrados no sistema.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Informação do Usuário.	O sistema mostra dados do usuário logado como: o login, nome, etc. Informações relacionadas a quem utiliza o sistema.
A6	É fácil ver (encontrar, localizar) a zona de Informação do Usuário.	É fácil enxergar e encontrar onde estão os dados do usuário logado.
A6	É fácil entender que os dados são do usuário logado no sistema.	Os dados disponibilizados indicam que usuário está usando o sistema no momento.
C6	É fácil entender se alguém está logado no sistema e quem ele é.	Por exemplo, caso não tenha ninguém logado o sistema mostra: "Entre". Se tiver alguém logado o sistema mostra: "Bem vindo: Fulano de Tal", ou "Você está logado como: fulano@email.com".
C6	É fácil entender as palavras/termos/símbolos usados para indicar quem está ou não logado no sistema.	Palavras como "Login", "Entrar" ou "Você está logado como:" são palavras que indicam se tem ou não um usuário logado. Estas palavras devem ser fáceis de entender.
C2	Para mostrar informação do usuário do sistema são usados termos e símbolos relacionados com o domínio da aplicação.	Por exemplo, Usuário: fulano@email.com.
A2	As informações do usuário são apresentadas de forma ordenada e lógica.	Por exemplo, num site de compras, os dados relevantes do usuário podem ser mostrados na ordem: E-mail, saldo disponível, pontos promocionais, etc.
A4	Os dados do usuário devem ser padronizados (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Ao usar um termo (como o e-mail para identificar o usuário) o mesmo deve ser usado ao longo da aplicação.
A8	A informação é legível.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.

Zona de Informação do Usuário		
Descrição		
Oferece ao usuário informações relacionadas ao usuário logado no sistema. Estas informações aparecem para usuários registrados no sistema.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A8	A informação é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento de informações importantes sobre o usuário logado no sistema.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Acesso Direto

Zona de Acesso Direto		
Descrição		
Contém funcionalidades Web comuns como: Login, Logout, Home, etc. Permite acesso direto a funcionalidades da aplicação que são mais usadas pelos usuários.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
7 - A7	Existe uma zona de Acesso Direto no Sistema.	Existem atalhos para acessar funcionalidades usadas pela maioria dos usuários ou de importância para o sistema. Por exemplo: Login, home, recados, etc.
A6	É fácil ver (encontrar, localizar) a zona de Acesso Direto no Sistema.	Os atalhos estão à vista para que os usuários possam vê-los
A6	É fácil entender que essa é a zona de Acesso Direto do Sistema.	É possível identificar os atalhos como tais, ou seja, é intuitivo que ao clicar em um o usuário é direcionado a uma parte funcional do sistema.
C6	É fácil entender as palavras/termos/símbolos usados para definir atalhos na zona de Acesso Direto.	Os atalhos usam termos que ajudam o usuário a identificar qual é a funcionalidade que será ativada.
C2	São usados termos e símbolos relacionados com o domínio da aplicação.	Por exemplo, o atalho para login é um botão de nome "Iniciar Seção". Os termos relacionados ao domínio da aplicação são de fácil entendimento para quem usa o sistema.
A2	Se houver atalhos: os atalhos se apresentam de forma ordenada e lógica.	Os atalhos não estão espalhados ou desorganizados.
A5	Se houver atalhos diferentes: É possível diferenciá-los.	Atalhos diferentes (Por exemplo, atalho de busca, atalho de login, etc.) possuem nomes/identificações diferentes.
A4	Os atalhos são padronizados (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Se um atalho é identificado com certo nome, símbolo ou cor, os mesmos devem ser mantidos para não confundir o usuário.
A7	Se houverem vários atalhos: sua disposição aumenta a eficiência de uso, minimizando o esforço de encontrá-los.	O usuário não passará muito tempo procurando ou identificando os atalhos.
N7	A interface permite ao usuário diferentes formas de acesso às tarefas	As tarefas de interesse do usuário devem ser apoiadas pelo sistema, disponibilizando meios de acesso

Zona de Acesso Direto		
Descrição		
Contém funcionalidades Web comuns como: Login, Logout, Home, etc. Permite acesso direto a funcionalidades da aplicação que são mais usadas pelos usuários.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	principais.	rápido e/ou caminhos alternativos.
A8	Os atalhos são legíveis.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	Os atalhos são úteis.	As informações são necessárias, não existem informações que atrapalham o uso dos atalhos.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de escolher um atalho errado:		
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de voltar ao estado anterior.
No caso de erro ao usar um atalho:		
A9	a) O sistema mostra uma mensagem de erro.	Se um atalho não funciona o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: Seu pedido não pode ser atendido no momento", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: Seu pedido não pode ser atendido no momento. Por favor, tente mais tarde ou entre em contato com o suporte fazendo clique aqui."

Itens de Verificação da Zona de Entrada de Dados

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Entrada de Dados no Sistema.	O sistema apresenta meios de inserção de dados: formulários, opções para selecionar itens, opções para marcar itens, ou meios de fornecer dados ao sistema a fim de realizar uma tarefa.
A6	É fácil ver (encontrar, localizar) onde entrar dados no sistema.	O usuário enxerga facilmente onde tem de informar dados.
A6	É fácil entender que essa zona é para entrar dados do Sistema.	Campos de preenchimento, seleção, etc. devem ser reconhecíveis, o usuário deve saber que eles servem para inserir dados.
C6	É fácil compreender como ingressar dados no sistema.	O meio utilizado para informar dados ao sistema deve ser intuitivo. Para preencher o nome, o formulário pode usar uma caixa de texto. Por exemplo para informar o sexo, a interface pode usar radio buttons "M" e "F".
C6	Avalie se é possível reconhecer o significado dos termos (palavras ou símbolos) utilizados.	O usuário deve conseguir identificar que dados estão sendo solicitados sem dificuldade. Por exemplo, um campo data, não devia possuir nomes ambíguos como "época", "tempo", etc.
A2	Os dados a serem informados pelo usuário (seja qual for o meio em que são fornecidos) estão apresentados em uma ordem natural e lógica segundo os conceitos do domínio do problema.	Ao realizar uma compra o usuário deve informar por exemplo: item a ser comprado, quantidade e forma de pagamento. A ordem de preenchimento dos dados deve ser lógica e segundo a funcionalidade sendo realizada.
C2	É usada a linguagem do domínio do problema.	Os termos usados devem ser facilmente entendíveis e relacionados com as funcionalidades a serem executadas no sistema.

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
C2	Os termos (palavras ou símbolos) utilizados seguem as convenções do mundo real.	Os termos usados devem fazer sentido para os usuários e não devem ser usados termos que sejam incomuns aos usuários. Uma compra não deve ser chamada de "investimento", pois nomes pouco usuais dificultam o entendimento do usuário.
A4	Há aderência a convenções de plataforma e padrões de interface adotados em relação a layout, formatação e controles.	Se uma data é preenchida no formato: dia/mês/ano, esse formato de preenchimento deve ser mantido e não alterado ao longo da aplicação.
A4	Tarefas similares possuem formulários de preenchimento similares. O estilo de preenchimento deve ser padronizado.	Ao informar dados em tarefas similares, os formulários não podem mudar radicalmente. Isso permite ao usuário saber como preencher dados em determinadas tarefas.
A5	Os dados obrigatórios na entrada de dados estão claramente definidos.	Por exemplo o sistema usa "*" ou "(obrigatório)" ao lado dos campos obrigatórios.
A7	Avalie se a interface permite a utilização de mecanismos de busca de informação que auxiliem a entrada de dados obrigatória.	Por exemplo, na hora de inserir um endereço, o sistema pode permitir procurar endereços no google maps.
A5	A interface indica o formato correto para uma entrada de dados específica.	Os campos devem conter/fornecer dicas de preenchimento. Ex. Data: dd/mm/aa, Telefone: 9999-9999.
A6-N10	A interface do sistema permite ao usuário visualizar informações sobre o formato correto de entrada de dados.	As dicas de preenchimento devem ser facilmente visualizadas pelo usuário.
C10	É possível compreender as informações sobre o formato correto de entrada de dados.	As dicas de preenchimento devem ser facilmente entendíveis.
C6	Dicas sobre os termos (palavras ou símbolos) específicos do domínio do problema estão acessível aos usuários.	O dado "Cod. de Seg" do cartão de crédito pode possuir a seguinte dica: "Código de três dígitos localizado no verso do cartão de crédito." Esta dica deve ser facilmente visualizada.

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
A6	É fácil reconhecer/visualizar dados já fornecidos.	Quando o usuário tiver informado um dado, o mesmo deve ficar disponível para visualização.
A5	Os dados a serem fornecidos são apresentadas de forma balanceada.	Os dados a serem fornecidos são solicitados em proporções adequadas. Não é necessário informar muitos dados de uma vez só.
A5	Se houver dados diferentes: a interface facilita a distinção entre os dados.	Em caso de serem solicitadas dois dados similares, cada um deve ser identificado. Por exemplo: Data de vencimento, data de pagamento.
A7	A disposição dos dados solicitados na interface do sistema aumenta a eficiência de uso.	Os dados são solicitados de forma ordenada e agrupados a fim de facilitar seu fornecimento.
N7	A interface permite ao usuário navegar com facilidade ao preencher os dados.	É fácil navegar dentro do mesmo formulário. Se o formulário é preenchido ao longo de várias páginas, deve ser fácil acessar as diferentes partes do formulário.
N7	A interface permite ao usuário a utilização de aceleradores ou atalhos na inserção de dados.	Por exemplo, um usuário pode salvar os dados do seu cartão de crédito para não preencher os mesmos dados a cada compra.
A8	Os campos a serem informados são legíveis.	A letra usada é fácil de ler com letras suficientemente grandes, bom contraste e fonte legível.
A8	Os campos a serem informados são úteis.	Os dados a serem informados são necessários, não existem informações que atrapalham o preenchimento dos dados. Dados desnecessários não são marcados como obrigatórios.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.
No caso de errar o preenchimento/seleção de um dado:		

Zona de Entrada de Dados		
Descrição		
Esta zona é responsável por providenciar ao usuário um meio de entrada de dados para executar determinadas operações. Posteriormente o usuário clicará em um botão que ativará uma função com base nos dados inseridos.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
N3	a) O sistema permite voltar.	Existem opções de “undo” ou formas de preencher o dado novamente.
	No caso de erro ao preencher/selecionar um dado:	
A9	a) O sistema mostra uma mensagem de erro.	Se o usuário não preencheu um dado obrigatório ou preencheu errado o sistema mostra uma mensagem.
A9	b) A mensagem é visível.	A mensagem usa fonte legível.
C9	c) A mensagem é entendível e usa linguagem simples.	Bom exemplo: "Erro: Data de nascimento é um campo obrigatório", Mal exemplo: "Erro 123".
C9	d) A mensagem sugere soluções alternativas.	Bom exemplo: "Erro: Data de nascimento é um campo obrigatório, por favor verifique se preencheu o dado e de forma correta"
A4	e) Avalie se as mensagens de erros são apresentadas de forma consistente com os padrões de apresentação utilizados.	Os erros similares devem ser notificados no mesmo padrão para o usuário identificar o tipo de erro cometido.

Itens de Verificação da Zona de Instituição

Zona de Instituição		
Descrição		
Oferece ao usuário informações sobre a Instituição (companhia, entidade, organização, etc.) que é responsável pela aplicação Web. Geralmente mostra informações sobre o nome da companhia, logo, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona que mostra a Instituição.	A página mostra o nome, logo, etc., da instituição responsável. Informações sobre quem é o dono do sistema ou o propósito geral do sistema.
A6	É fácil ver (encontrar, localizar) a zona que mostra dados da Instituição.	O usuário consegue enxergar o nome/logo da instituição.
A6	É fácil entender que essa é a zona que mostra dados da Instituição.	O usuário deve reconhecer que as imagens/palavras mostradas são referentes à instituição.
C6	É fácil entender qual é a instituição do sistema.	O logo/nome deve ser sugestivo e o usuário deve entender qual é o propósito do site.
C6	É fácil entender as palavras/termos/símbolos usados para indicar qual é a instituição do sistema.	É fácil entender a mensagem passada no logo, nome, etc. e o que ele representa.
A4	As informações sobre a instituição são apresentadas de forma padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da aplicação.	Se um logo é usado em uma página o mesmo deve ser mantido nas outras. Manter consistência e padrões.
A8	A informação sobre a instituição é legível.	A letra/imagem usada é fácil de ler/ver com letras suficientemente grandes, bom contraste e fonte legível.
A8	A informação sobre a instituição é útil.	As informações são necessárias, não existem informações que atrapalham o entendimento do usuário.
A8	O design é agradável (gosto dele).	As cores usadas são usadas corretamente. Não existem cores fortes que causam desconforto físico ou distraem.

Itens de Verificação da Zona de Customização

Zona de Customização		
Descrição		
Contém informações sobre outros conteúdos que não podem ser classificados dentro das zonas anteriormente descritas. Geralmente é usada para mostrar conteúdo que não está relacionado ao domínio da aplicação como: propagandas, outras aplicações Web, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	Existe uma zona de Customização no Sistema (Ex. Conteúdo não relacionado com o domínio da aplicação, aplicações externas, propaganda, etc.).	
A6	É fácil ver (encontrar, localizar) a zona de Customização.	
A6	É fácil entender que essa é a zona de Customização.	
C6	É fácil compreender o que pode ser feito na zona de Customização.	
C6	É fácil entender as palavras/termos/símbolos usados na zona de Customização.	
C2	Os termos e símbolos relacionados com o objetivo da zona de Customização (Ex. Uma propaganda apresentada faz sentido).	
A2	Se forem apresentados componentes (botões, imagens, texto): os componentes da zona de Customização se apresentam de forma lógica e ordenada.	
A5	Se forem apresentados componentes (botões, imagens, texto): É possível diferenciar entre os possíveis componentes e serviços da zona de Customização.	
7	Se for necessário um atalho para agilizar a interação: o sistema disponibiliza opções para acelerar a interação na zona de customização.	
A4	A zona de Customização é padronizada (existem cores, layout, símbolos e termos estabelecidos) ao longo da	

Zona de Customização		
Descrição		
Contém informações sobre outros conteúdos que não podem ser classificados dentro das zonas anteriormente descritas. Geralmente é usada para mostrar conteúdo que não está relacionado ao domínio da aplicação como: propagandas, outras aplicações Web, etc.		
Base Teórica	Itens de Verificação	Exemplo/Esclarecimento
	aplicação.	
	O design da zona de Customização:	
8	a) É agradável (Gosto dele).	
A8	b) Não produz desconforto físico. (Ex. o uso de cores fortes, ou letras pequenas e ilegíveis).	
A8	c) É simples (Não existem informações que atrapalham o entendimento).	
	Caso ocorra um erro (Ex. tomar um caminho errado):	
9	a) O usuário conseguirá identificar que errou.	
9	b) O usuário conseguirá dizer que tipo de erro foi. (Ex. Click errado, escolher menu errado, etc.)	
3	c) O usuário poderá desfazê-lo.	
9	d) O usuário poderá corrigi-lo.	