Método Baseado em Heurísticas para Avaliação de Acessibilidade em Sistemas de Informação

Este exemplar corresponde à redação final da Tese devidamente corrigida e defendida por Eduardo Hideki Tanaka e aprovada pela Banca Examinadora

Campinas, 11 de fevereiro de 2010.

Heloísa Vieira da Rocha (Orientadora)

Tese apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP
Bibliotecária: Crisllene Queiroz Custódio – CRB8 / 7966

Tanaka, Eduardo Hideki

T153m Método baseado em heurísticas para avaliação de acessibilidade em sistemas de informação / Eduardo Hideki Tanaka -- Campinas, [S.P. : s.n.], 2009.

Orientadora : Heloísa Vieira da Rocha

Tese (Doutorado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Acessibilidade. 2. Design universal. 3. Avaliação. 4. Heurística. 5. Usabilidade. I. Rocha, Heloísa Vieira da. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Título em inglês: Method based on heuristics to accessibility evaluation in information systems

Palavras-chave em inglês (Keywords): 1. Accessibility. 2. Universal design. 3. Evaluation. 4. Heuristic. 5. Usability.

Área de concentração: Sistemas de Informação

Titulação: Doutor em Ciência da Computação

Banca examinadora: Profa. Dra. Heloísa Vieira da Rocha (IC-Unicamp)
Profa. Dra. Joice Lee Otsuka (Universidade Aberta do Brasil - UFSCar)
Prof. Dr. Delano Medeiros Beder (EACH-USP Leste)
Profa. Dra. Fernanda Maria Pereira Freire (NIED-UNICAMP)
Prof. Dr. Rodolfo Jardim de Azevedo (IC-UNICAMP)

Data da defesa: 22/12/2009

Programa de Pós-Graduação: Doutorado em Ciência da Computação

TERMO DE APROVAÇÃO

Tese Defendida e Aprovada em 22 de dezembro de 2009, pela Banca examinadora composta pelos Professores Doutores:

Profª. Drª. Joice Lee Otsuka
UABrasil / UFSCar

Prof. Dr. Delano Medeiros Beder
EACH / USP

Profª. Drª. Fernanda Maria Pereira Freire
NIED/ IEL / UNICAMP

Prof. Dr. Rodolfo Jardim de Azevedo
IC / UNICAMP

Profª. Drª. Heloísa Vieira da Rocha
IC / UNICAMP

Instituto de Computação
Universidade Estadual de Campinas

Método Baseado em Heurísticas para Avaliação de Acessibilidade em Sistemas de Informação

Eduardo Hideki Tanaka

Dezembro de 2009

Banca Examinadora:

- Profª Drª Heloísa Vieira da Rocha (Orientadora)
  Instituto de Computação – UNICAMP
- Profª Drª Joice Lee Otsuka
  Universidade Aberta do Brasil – UFSCar
- Prof. Dr. Delano Medeiros Beder
  Escola de Artes, Ciências e Humanidades – USP - Leste
- Profª Drª Fernanda Maria Pereira Freire
  Núcleo de Informática Aplicada à Educação/Instituto de Estudos da Linguagem – UNICAMP
- Prof. Dr. Rodolfo Jardim de Azevedo
  Instituto de Computação – UNICAMP
- Prof. Dr. Marcelo Morandini (Suplente)
  Escola de Artes, Ciências e Humanidades – USP - Leste
- Prof. Dr. Ivan Luiz Marques Ricarte (Suplente)
  Faculdade de Engenharia Elétrica e de Computação – UNICAMP

Resumo

Para ser mais inclusivo e menos exclusivo, o design de sistemas de informação deve considerar questões de acessibilidade. Atualmente, existem padrões para orientar o design acessível e, também, alguns métodos para avaliação de acessibilidade, voltados principalmente para conteúdo na Web. Dos métodos de avaliação de acessibilidade existentes, os mais adotados hoje em dia são aqueles baseados na verificação de conformidade com guias de design acessível e os testes com usuários com deficiências. Tanto um quanto o outro podem ser considerados métodos caros e que demandam um tempo considerável para aplicação e análise, sendo esta uma das razões para que muitos desenvolvedores simplesmente ignorem a avaliação da acessibilidade durante o ciclo de desenvolvimento de software. Sabendo disto, esta tese de doutorado propõe exatamente um método alternativo para avaliar a acessibilidade em sistemas de informação, baseado em heurísticas. Os resultados obtidos através de dois experimentos mostraram que as heurísticas de acessibilidade propostas são fáceis de aprender, rápidas de aplicar e de baixo custo, o que possibilita sua aplicação a qualquer momento do processo de desenvolvimento de um software.

Abstract

To be more inclusive and less exclusive, the design of information systems must take into account accessibility issues. Nowadays, there are standards to guide the accessible design and, also, some accessibility evaluation methods, focused on the assessment of Web content, mainly. From all accessibility evaluation methods available, the most adopted nowadays are guidelines review and tests with users with disabilities. Both of them can be considered expensive methods and require a reasonable time to apply and to analyze so that several developers simply ignore accessibility evaluation during the software development cycle. Knowing these issues, this PhD thesis proposes an alternative method to evaluate accessibility, based on heuristics. The results of two experiments showed that the proposed accessibility heuristics are easy to learn, fast to apply and not expensive, therefore, it could be applied anytime during the software development cycle.

Publicações

Freire, A. P.; Power, C.; Petrie, H; Tanaka, E. H.; Rocha, H. V.; Fortes, R. P.M. (2009) Web accessibility metrics: effects of different computational approaches. In: Proceedings of the 13th International Conference on Human-Computer Interaction (HCI International 2009), jul/2009, San Diego, California, USA.

Freire, A. P.; Tanaka, E. H.; Lara, S. M. A.; Rocha, H. V.; Fortes, R. P. M. (2007) Comparing Accessibility in e-learning Web environments: a metric based approach. In: Proceedings of the XIII Brazilian Symposium on Multimedia and the Web (Webmedia 2007), Gramado/RS.

Tanaka, E. H.; Vargas, A.; Silva, A. C.; Rocha, H. V. (2006) Centenas de razões para achar o JEMS difícil. In: Anais do VII Simpósio sobre Fatores Humanos em Sistemas Computacionais (IHC 2006), Competição de Avaliação de Sistema, Natal/RN.

Tanaka, E. H.; Bim, S. A.; Rocha, H. V. (2005) Comparing accessibility evaluation and usability evaluation in HagáQuê. In: Proceedings of the 2nd Latin American Conference on Human-Computer Interaction (CLIHC 2005), Cuernavaca, Mexico.

Tanaka, E. H.; Juliato, M.; Baudet, C.; Galves, M.; Coelho, T. T.; Rocha, H. V. (2005) Desenvolvimento de um teclado virtual com diversos layouts e varreduras. In: Anais do V Congresso Ibero-americano de Informática na Educação Especial (CIIEE2005), Montevidéu, Uruguai.

Juliato, M.; Tanaka, E. H.; Baudet, C.; Galves, M.; Coelho, T. T.; Rocha, H. V. (2004) TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura. In: Anais do VI Simpósio de Fatores Humanos em Sistemas Computacionais (IHC2004), Curitiba/PR.

Juliato, M.; Tanaka, E. H.; Baudet, C.; Galves, M.; Coelho, T. T.; Rocha, H. V. (2004) TFlex: Proposta de Simulador de Teclado com Diferentes Varreduras e Layouts, In: Anais do VII Congresso Iberoamericano de Informática Educativa (RIBIE2004), Monterrey, Mexico.

Agradecimentos

Aos meus pais e aos meus irmãos, presentes em todos os momentos, sem os quais eu não chegaria até aqui.

À minha grande amiga e orientadora, Heloísa, por todas as lições aprendidas.

Aos professores Joice, Delano, Fernanda, Rodolfo, Marcelo e Ivan, por aceitarem o convite para compor minha banca de defesa de doutorado, mesmo sendo na véspera de Natal.

À Patrícia, amor da minha vida, pelo seu carinho.

Aos meus amigos Alexandre, Daniel, Cláudio, Rodrigo e Wagner, há mais de 15 anos compartilhando alegrias.

Aos também amigos Andressa, André, Ariel, Fernanda, Mariana, Marcio, Ricardo, Rodrigo, Selma, Sílvia e tantos outros companheiros de vida acadêmica.

A todos os participantes dos testes e experimentos e suas fundamentais contribuições para a consolidação dessa tese.

Aos meus ex-alunos, pelas discussões, pela paciência e motivação em aprender alguma coisa mesmo depois das 21 horas.

Ao Instituto de Computação e ao Núcleo de Informática Aplicada à Educação da UNICAMP, por terem acolhido o meu trabalho acadêmico de doutorado.

À CAPES, ao CNPq e à FAPESP, pelo financiamento de minha pesquisa de doutorado.

E a você, meu caro leitor, pelo interesse em minha tese.

Sumário

Resumo vii

Abstract ix

Publicações xi

Agradecimentos xiii

Índice de Figuras xix

Índice de Tabelas xxi

1 Introdução 23

1.1. Heurística vs. guideline 25

1.1.1. O que é heurística? 26

1.1.2. Uma Heurística é uma *Guideline*, e vice-versa? 27

1.1.3. De onde vêm as heurísticas de IHC? 27

2 Usabilidade 31

2.1. Definição 31

2.2. Métodos de avaliação de usabilidade 34

2.2.1. Avaliação Heurística 35

2.2.2. Percurso Cognitivo 38

2.2.3. Testes com usuários 39

3 Acessibilidade e Design Universal 45

3.1. Pessoas com Deficiências 45

3.2. Origens históricas da acessibilidade e do design universal 47

3.3. Princípios do Design Universal 50

3.4. Acessibilidade em sistemas de informação 53
3.3.1. Tecnologias assistivas 58
3.5. Guias de acessibilidade 63
3.4.1. WCAG 1.0 63
3.4.2. ATAG 1.0 65
3.4.3. UAAG 1.0 66
3.4.4. WCAG 2.0 67
3.4.5. Por que as WCAG 2.0 ainda não deslancharam? 68
3.4.6. Por que o modelo de acessibilidade da W3C é falho? 70
3.4.7. Desafios ao lecionar sobre as WCAG 73
3.4.8. Outras *guidelines* e padrões de acessibilidade 76
3.4.8.1. *Guidelines* para Adobe Flash 76
3.4.8.2. *Guidelines* para Java 77
3.4.8.3. *Guidelines* IBM 77
3.4.8.4. *Guidelines* GNOME para GNU/Linux 78
3.6. Leis de acessibilidade 79
3.7. Avaliação de Acessibilidade 80
3.6.1. Conformidade com *guidelines* 80
3.6.2. Por que a avaliação de acessibilidade com as WCAG nem sempre é apropriada? ... 81
3.6.3. Estudo sobre as ferramentas semi-automáticas 82
3.6.3.1. Resultados do estudo sobre as ferramentas semi-automáticas 88
3.6.4. Testes com usuários 93
3.6.5. Percurso de barreiras 94
3.6.6. Percurso cognitivo estendido 96
3.6.7. Heurísticas de acessibilidade já existentes 96
4 Processo de elaboração das heurísticas 99

4.1. Fase de extração 101
4.2. Fase de refinamento 104
4.2.1. Primeiro Experimento 106
4.2.1.1. Materiais e métodos 106
4.2.1.2. Resultados 108
4.2.2. Segundo experimento 120
4.2.2.1. Materiais e métodos 122
4.2.2.2. Resultados 123
4.3. Versão final das heurísticas de acessibilidade 132
5 Conclusões 135
Referências 139
Anexo A – Materiais do experimento com ferramentas de acessibilidade 151
Anexo B – Interfaces exploradas na fase de extração 159
B.1. Microscópio Virtual 159
B.2. Quebra-Cabeça 161
B.3. Jogo da Reciclagem 162
B.4. Cada Caso é um Caso 164
B.5. Geometria 165
B.6. Não me sinto bem 167
B.7. GeoNet 168
B.8. Pre-Season Puzzles 169
B.9. Pidgin 171
Anexo C – Materiais usados na fase de refinamento 173
Anexo D – Resultados da avaliação de acessibilidade – primeiro experimento 191
Anexo E – Resultados da avaliação de acessibilidade – segundo experimento 195
Anexo F – Heurísticas de acessibilidade – versão final 199

Índice de Figuras

Figura 2. 1. Modelo de aceitabilidade de um sistema. 32

Figura 2. 2. Relação entre número de participantes e a quantidade de problemas de usabilidade encontrados em testes com usuários (Nielsen, 2000). 41

Figura 3. 1. Exemplos de acionadores para uso com teclados virtuais. À esquerda, botões circulares de diferentes tamanhos. Ao centro, um pedal. À direita, acionador baseado em sopro e sucção..... 60

Figura 3. 2. Teclado Virtual do Windows XP. No caso, a varredura linear está passando exatamente pela letra "j" (em destaque). 60

Figura 3. 3. Escrita com o Dasher. 61

Figura 3. 4. Exemplo de teclados expandido (à esquerda) e reduzido (à direita)..... 62

Figura 3. 5. Relação dos alicerces de acessibilidade da W3C. 70

Figura 3. 6. Página inicial do MEC. No detalhe ampliado, no canto inferior esquerdo, o selo "AAA" apresentado na parte de baixo da página. 75

Figura 3. 7. Relatório textual da ferramenta Truwex. 83

Figura 3. 8. Relatório gráfico da ferramenta Truwex, indicando o local na página onde há uma violação de *checkpoint* das WCAG 1.0. 83

Figura 3. 9. Relatório de violações de *checkpoints* da ferramenta DaSilva. 84

Figura 3. 10. Relatório de acessibilidade da ferramenta HTML Validator for Firefox..... 85

Figura 3. 11. Relatório gerado pela ferramenta Fujitsu Web Accessibility Inspector..... 86

Figura 4. 1. Processo de elaboração das heurísticas de acessibilidade..... 100

Figura 4. 2. Respostas dos avaliadores para as perguntas sobre os tempos para realizar uma avaliação e para o aprendizado das heurísticas (primeiro experimento). 116

Figura 4. 3. Respostas dos avaliadores para as perguntas sobre dúvidas iniciais e a diminuição das mesmas com o avanço da avaliação (primeiro experimento)..... 116

Figura 4. 4. Respostas dos avaliadores para as perguntas sobre a facilidade de aplicação das heurísticas comparadas às *guidelines* da W3C e sobre a facilidade do material de apoio cedido (primeiro experimento). 116

Figura 4. 5. Respostas dos avaliadores para as perguntas sobre o entendimento das heurísticas de acessibilidade (primeiro experimento). 117

Figura 4. 6. Respostas dos avaliadores para as perguntas sobre os problemas encontrados (primeiro experimento). 117

Figura 4. 7. Respostas dos avaliadores para as perguntas sobre a possibilidade de novo uso das heurísticas e a satisfação com os resultados obtidos (primeiro experimento). 118

Figura 4. 8. Respostas dos avaliadores para as perguntas sobre a possibilidade de se melhorar as avaliações à medida que se torna mais experiente (primeiro experimento). 118

Figura 4. 9. Respostas dos avaliadores para as perguntas sobre o tempo para realizar uma avaliação e clareza das heurísticas (segundo experimento). 129

Figura 4. 10. Respostas dos avaliadores para as perguntas sobre o entendimento das heurísticas de acessibilidade (segundo experimento). 129

Figura 4. 11. Respostas dos avaliadores para as perguntas sobre os problemas encontrados (segundo experimento). 130

Figura 4. 12. Respostas dos avaliadores para as perguntas sobre a possibilidade de novo uso das heurísticas e a satisfação com os resultados obtidos (segundo experimento). 130

Figura 4. 13. Respostas dos avaliadores para as perguntas sobre a possibilidade de se melhorar as avaliações à medida que se torna mais experiente (segundo experimento). 130

Figura B. 1. Lâmina sendo analisada no Microscópio Virtual. 160

Figura B. 2. Jogo do Quebra-Cabeça, com parte dos Estados fora de posição. 162

Figura B. 3. Escolha de lixeira no Jogo da Reciclagem. 163

Figura B. 4. Escolha de analgésico no Cada Caso é um Caso. 164

Figura B. 5. Análise de um poliedro no software Geometria. 166

Figura B. 6. Não me sinto bem, quando a doença mostrada é a malária. 167

Figura B. 7. Perguntas sobre os países da América do Sul no GeoNet. 168

Figura B. 8. Palavras cruzadas no Pre-Season Puzzles. 170

Figura B. 9. Troca de mensagens no Pidgin. 171

Índice de Tabelas

Tabela 2. 1. Versão revisada das heurísticas de Nielsen. 35

Tabela 2. 2. Graus de severidade..... 36

Tabela 3. 1. População brasileira com deficiência (IBGE, 2000). 56

Tabela 3. 2. Ordem de testes das ferramentas, por participante. 87

Tabela 3. 3. Páginas analisadas com cada ferramenta..... 88

Tabela 3. 4. Exemplo de barreira segundo Brajnik (2006). 95

Tabela 4. 1. Problemas encontrados para cada uma das heurísticas no primeiro experimento..... 111

Tabela 4. 2. Problemas encontrados por participante no primeiro experimento. 112

Tabela 4. 3. Resultados dos questionários do primeiro experimento (média, desvio padrão e moda). 115

Tabela 4. 4. Problemas encontrados para cada uma das heurísticas no segundo experimento. 125

Tabela 4. 5. Problemas encontrados por participante no segundo experimento..... 126

Tabela 4. 6. Resultados dos questionários do segundo experimento (média, desvio padrão e moda). 128

Tabela D. 1. Problemas encontrados durante o primeiro experimento com as heurísticas de acessibilidade. 191

Tabela E. 1. Problemas encontrados durante o segundo experimento com as heurísticas de acessibilidade. 195

Capítulo 1

Introdução

O tema acessibilidade é extremamente intimidador para muitos designers e desenvolvedores de software. De fato, há questões complexas ao se tratar de acessibilidade e alguns designers e desenvolvedores não sabem como lidar com elas. Adicionalmente, faltam bons recursos para orientar o desenvolvimento de sistemas de informação usáveis por quaisquer pessoas, independente de terem ou não algum tipo de deficiência (Choi et al., 2006).

Entretanto, e ao mesmo tempo, há uma pressão cada vez maior por parte da sociedade e razões políticas, econômicas, éticas e morais para que todos os recursos tecnológicos se tornem mais acessíveis (Theofanos e Redish, 2003). Não há dúvida, portanto, de que algo deve ser feito para que mundo se torne **menos exclusivo e mais inclusivo**.

Todas estas dificuldades e pressões acabaram sendo vivenciadas por mim, autor desta tese de doutorado, ao longo dos últimos anos. Desde o ano 2000, trabalho em projetos de software nos quais a acessibilidade tem um papel importante. O primeiro destes projetos foi o do software HagáQuê (2001).

O HagáQuê é um software livre, desenvolvido como parte da dissertação de mestrado de Bim (2001) e da minha iniciação científica, ainda durante minha graduação (Bim et al., 2000). Trata-se de uma ferramenta de autoria de histórias em quadrinhos para estimular o aprendizado e o uso da língua escrita por crianças nas fases iniciais de alfabetização e sem experiência no uso de computadores. No entanto, no projeto inicial do HagáQuê, a acessibilidade não foi considerada como um requisito de qualidade, tampouco foram incluídas como público-alvo crianças com deficiências. Por conta disto, diversos problemas de acessibilidade foram descobertos logo no primeiro uso do HagáQuê por

crianças com deficiências – e isto ocorreu justamente em um projeto que envolvia centenas de escolas de educação especial em todo o Brasil – e, consequentemente, centenas de alunos com deficiências (Proinesp, 2001)!

A necessidade de um design mais acessível ficou evidente no HagáQuê e motivou a minha dissertação de mestrado (Tanaka, 2004). Porém, todas as avaliações de acessibilidade do HagáQuê durante este processo de (re)design foram feitas através de testes que envolveram dezenas de crianças com as mais diversas deficiências e transtornos (visão subnormal, Síndrome de Down, autismo, Síndrome de Asperger, surdez, deficiências motoras, entre outras). Isto porque os métodos existentes de inspeção de acessibilidade, teoricamente mais rápidos e baratos que os testes com usuários exatamente por não exigirem a participação de usuários, eram destinados a avaliar apenas sistemas Web, o que não é o caso do HagáQuê. Apesar dos testes com usuários serem considerados na literatura como fundamentais e gerarem resultados mais objetivos e precisos que os métodos de inspeção (Law e Hvannberg, 2002), são muito mais caros e demorados, o que pode comprometer o orçamento e o cronograma de qualquer projeto de desenvolvimento de software.

No parágrafo acima, vale a pena destacar a afirmação de que os métodos de inspeção de acessibilidade existentes são **teoricamente** mais rápidos e baratos que os testes com usuários. Tal afirmação é verdadeira quando se trata de avaliar usabilidade apenas. Com relação à acessibilidade, o principal método de inspeção atualmente é a verificação de conformidade com *guidelines*, que procura encontrar potenciais problemas de acessibilidade especialmente em conteúdo de páginas Web. Mesmo ignorando a limitação no escopo da avaliação, há outros problemas tão ou mais severos com este método de avaliação de acessibilidade.

Alguns estudos demonstram que, na prática, a avaliação da acessibilidade através de verificação de conformidade com *guidelines* é um método custoso, complexo e demorado. De fato, em um estudo realizado por mim e que será discutido mais adiante nesta tese de doutorado, foi constatado que mesmo usando-se algumas ferramentas auxiliares existentes para verificar a conformidade com *guidelines*, é preciso uma hora em média para avaliar uma única página Web.

Outros problemas relacionados com a verificação de conformidade com *guidelines* têm relação com o conjunto de *guidelines* em si. Tipicamente, são adotadas as *Web Content Accessibility Guidelines 1.0* (WCAG 1.0) (Chisholm et al., 2001). As WCAG 1.0 foram desenvolvidas pela W3C e se tornaram um padrão de acessibilidade para a Web no mundo, inspirando inclusive outros conjuntos de *guidelines*. As WCAG 1.0 e outros conjuntos de *guidelines* inspirados nelas são duramente criticados e contestados e uma das principais razões é justamente por não terem sido concebidos pensando-se nos desenvolvedores e designers que os usariam (Kelly et al., 2007; Choi et al., 2006).

Com base nestas e outras experiências vivenciadas, a presente tese de doutorado objetivou exatamente a concepção de um método alternativo de avaliação de acessibilidade. Da mesma forma como Nielsen (1994) concebeu um conjunto de heurísticas de usabilidade, um método de inspeção de usabilidade comprovadamente barato, rápido de aplicar e fácil de aprender, a proposta contida nesta tese é a de um conjunto de heurísticas de acessibilidade, que visa ter as mesmas características das heurísticas de Nielsen (barato, rápido de aplicar e fácil de aprender).

Os próximos capítulos apresentam as justificativas, o embasamento teórico e o desenvolvimento prático desta tese. O Capítulo 2 discorre a respeito do conceito de usabilidade e os principais métodos para avaliação de usabilidade. No Capítulo 3, os temas centrais são acessibilidade e design universal, permeando também os principais métodos existentes para avaliação de acessibilidade, suas principais vantagens e desvantagens. O Capítulo 4 é destinado a descrever detalhadamente o processo de desenvolvimento das heurísticas de acessibilidade e os resultados que foram obtidos. Por fim, as discussões sobre as principais descobertas e as considerações finais estão no Capítulo 5.

1.1. Heurística vs. guideline

Ao longo dos próximos capítulos, diversos conceitos essenciais para o bom entendimento desta tese de doutorado serão apresentados: usabilidade, design universal, acessibilidade, métodos de inspeção e métodos empíricos para avaliação de interfaces, entre

outros. No entanto, dois conceitos igualmente relevantes merecem um destaque logo no início: heurística e *guideline*.

Ambos os conceitos são muito próximos e vale a pena uma análise sobre suas semelhanças e diferenças.

1.1.1. O que é heurística?

Segundo a Wikipedia (2009a), heurística é uma técnica baseada na experiência que ajuda na resolução de problemas. Um método heurístico é usado para se chegar rapidamente a uma solução próxima da melhor possível, ou seja, a uma solução ótima. É uma *rule of thumb*, um palpite, um julgamento intuitivo ou, simplesmente, senso comum.

Já pelo Dictionary.com (2009a), pode-se definir heurística dentro da área de computação como uma simplificação, um bom palpite que reduz ou limita a busca por soluções em domínios que são difíceis ou não são muito bem compreendidos.

Outro dicionário livre, o Free Dictionary (2009a) cita a origem da palavra heurística, que vem do grego *heuriskein*, que significa descobrir; é uma regra ou conjunto de regras definidas para aumentar a probabilidade de resolver algum problema.

Puchkin (1976), cientista russo, descreve heurística de uma forma mais abrangente, referenciando vários fatos históricos desde a Grécia Antiga. Para Puchkin, a atividade heurística é o pensamento criador de uma nova estratégia, inédita, para resolver um problema, que não foi obtida unicamente através do raciocínio lógico, muito pelo contrário. Segundo Puchkin, a intuição tem uma contribuição considerável neste processo: "o homem chega à resposta que procura sem ter consciência do processo pelo qual ela foi atingida" (Puchkin, 1976, p. 13).

Um dos referenciados por Puchkin é exatamente o matemático húngaro Pólya, que dedicou parte de sua vida à caracterização dos procedimentos que as pessoas adotam para resolver problemas de todos os tipos, especialmente os matemáticos. Para Pólya (1975), heurística trata de estudos de regras e métodos que auxiliam a se chegar a descobertas e invenções – e, como adjetivo, heurística significa "servir para descobrir".

Portanto, seguindo estes conceitos de heurística, pode-se dizer que as heurísticas de usabilidade de Nielsen (1994) são um conjunto de regras para auxiliar na descoberta de problemas de usabilidade em uma interface. Ou seja, o problema que Nielsen procura resolver com suas heurísticas é exatamente encontrar problemas de usabilidade em uma interface.

1.1.2. Uma Heurística é uma *Guideline*, e vice-versa?

Apesar de similares e muitas vezes serem usadas como sinônimos, heurística e *guideline* não são exatamente a mesma coisa, haja vista suas definições. Uma *guideline*, ou linha guia, é qualquer documento que objetiva adequar processos específicos. Ainda, por definição, seguir uma *guideline* não é mandatório (Wikipedia, 2009b).

No Dictionary.com (2009b), *guideline* é definida como algo para guiar os passos ou movimentos de alguém. E no Free Dictionary (2009b), *guideline* é uma sentença ou princípio para propor um conjunto de padrões ou determinar um curso de ações. Ainda, outra definição contida no Free Dictionary é a de que *guideline* é uma regra ou princípio que fornece uma guia para um comportamento considerado apropriado.

A partir das definições de heurística e *guideline*, pode-se dizer que ambas podem ser consideradas regras. Todavia, com objetivos bem distintos. Enquanto heurística é uma regra que ajuda a resolver um determinado problema, *guideline* é uma regra estabelecida para padronizar ou uniformizar um determinado comportamento esperado.

1.1.3. De onde vêm as heurísticas de IHC?

As definições de heurística apresentadas anteriormente são genéricas e podem sugerir que elas são fruto da intuição individual de seus autores. Entretanto, nem sempre as

heurísticas são resultado de uma mente inspirada e inexplicavelmente sem qualquer relação com uma pesquisa científica e o raciocínio lógico.

Chisnell (2009), pesquisadora de como pessoas da terceira idade interagem com a tecnologia, escreveu em seu blog um artigo exatamente sobre como as heurísticas da área de IHC (Interação Humano-Computador) são criadas. Para ela, há três grandes fontes principais: conhecimento ou sabedoria popular, experiência de especialistas e pesquisa.

Heurísticas vindas de conhecimento ou sabedoria popular seriam aquelas que ninguém exatamente sabe como surgiram, ou se efetivamente trazem algum efeito na resolução de um problema. Adicionalmente, em alguns casos, parece não existir sequer um relacionamento lógico aparente entre o problema e a heurística. Por exemplo, dizer que para se melhorar de um resfriado, a pessoa deve comer muito e para melhorar de uma febre deve comer pouco (Chisnell, 2009).

Na área de Web design, Chisnell (2009) cita outro exemplo. Uma regra popular entre Web designers é a de que o número de itens em um menu de navegação de um site deveria ser limitado entre 5 e 7. A suposição é a de que esta regra foi cunhada a partir da pesquisa de Miller (1956) sobre o limite da memória de trabalho do ser humano (ou memória de curta duração), estimado entre 7 mais ou menos 2 blocos de informação, ou "chunks". Contudo, como Chisnell argumenta, não faz sentido imaginar que o usuário deve ter sempre todos os itens de um menu de navegação em sua memória de trabalho a cada página visitada; a navegação é persistente e o menu de navegação existe exatamente para que o usuário não precise mantê-lo em sua memória de trabalho. Ou seja, seguir este tipo de regra popular não garante uma boa usabilidade em um site. Um estudo com os usuários-alvo sobre quais itens de navegação são mais relevantes para eles seria muito mais adequado.

A segunda fonte de heurísticas é a experiência de especialistas, segundo Chisnell (2009). E apenas observando as mesmas coisas ocorrerem com os mesmos tipos de usuários inúmeras vezes é que se adquire o *expertise* necessário para se interiorizar este tipo de heurística. Chisnell cita sua própria experiência com pessoas da terceira idade, na qual chegou a princípios como: "Adultos mais velhos que usam a Web necessitam de alvos grandes e com alto contraste". Chisnell tem a certeza de que, seguindo este e outros princípios que ela tem em sua mente, o design de um site se torna muito mais fácil de usar

por pessoas da terceira idade. Contudo, justificar avaliações de interfaces a partir apenas de opiniões ou "pistas" que o especialista possui pode ser um tanto quanto difícil.

Por fim, Chisnell (2009) afirma que a melhor forma de se obter heurísticas é por meio de pesquisa de estudos já publicados e, novamente, justifica a partir de seu trabalho no site usability.gov (2006), no qual identificou e analisou mais de 500 princípios de design. A análise se deu a partir de estudos previamente publicados, que foram analisados por uma equipe de revisores constituída por especialistas de usabilidade e por Web designers. Os revisores avaliaram a "importância relativa", que indica o "quão importante é este princípio para o sucesso de um Web site", e a "força de evidência", que indica a credibilidade e a validade dos princípios – se, por exemplo, ao menos um estudo formal e rigoroso que o valida existe, resultando em um extenso conjunto de princípios. Adicionalmente, para Chisnell (2009), heurísticas concebidas a partir deste tipo de pesquisa acabam por possuir mais credibilidade, especificidade e aplicabilidade que os dois modelos de obtenção de heurísticas anteriormente apresentados.

Além de Chisnell, Paddison e Englefield (2003) também descreveram o processo de desenvolvimento de heurísticas de IHC. Para eles, há dois métodos práticos para se criar um conjunto de heurísticas. Um deles é exatamente o método baseado em pesquisa citado por Chisnell. O outro é o método baseado em avaliação, no qual se deve analisar os resultados de avaliações prévias de interfaces e extrair as principais descobertas como heurísticas.

Somervell et al. (2003) são outros pesquisadores que propuseram heurísticas, no caso, para sistemas de exibição de informações em telas grandes. Em sua pesquisa, Somervell et al. utilizaram uma abordagem diferente para obtenção de heurísticas, partindo de técnicas do design baseado em cenários (Rosson & Carroll, 2002). Cinco sistemas existentes de exibição de informações em telas grandes foram selecionados e inspecionados por Somervell et al., que, então, levantaram assertivas relacionadas ao que os usuários podem ou não fazer com estes sistemas, como a informação é mostrada e como o usuário deveria interagir. Um exemplo de assertiva descoberta é:

> Utilizar piscar (*blink*) para destacar novos itens pode:
> + oferecer rápido reconhecimento de novas informações
> - mas pode aumentar a distração de outras tarefas

Após o levantamento inicial de assertivas para os sistemas inspecionados, os autores discutiram e debateram sobre como elas impactam nos objetivos dos usuários, procurando relacioná-las e agrupá-las para, então, sintetizá-las em heurísticas de alto nível (Somervell et al., 2003).

Assim, resumidamente, pode-se dizer que há pelo menos cinco métodos para a obtenção de heurísticas de IHC:

- baseado em conhecimento popular
- baseado em experiências de especialistas
- baseado em avaliações
- baseado em assertivas de design baseado em cenários
- baseado em pesquisa

E, independente do método adotado para obtenção de um conjunto de heurísticas de usabilidade ou acessibilidade, é crucial a realização de algum tipo de avaliação do conjunto de heurísticas, para que se identifique, por exemplo, se as heurísticas são fáceis de aprender, fáceis de memorizar e se são expressivas no sentido de que os avaliadores conseguem sempre associar um problema na interface a pelo menos uma das heurísticas (Paddison e Englefield, 2003).

Capítulo 2

Usabilidade

Este capítulo trata do conceito de usabilidade e os métodos de avaliação de usabilidade em sistemas de informação.

2.1. Definição

Usabilidade não é uma palavra estranha entre os designers da atualidade. A norma ISO 9241-11 (ISO, 1998), que orienta sobre como a usabilidade de um produto pode ser especificada e avaliada, define usabilidade como "a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico". Ou seja, a usabilidade depende do contexto de uso, que é composto por usuários, tarefas, equipamentos (hardware e software) e do ambiente físico e social.

Para Nielsen (1993), renomado autor de livros sobre engenharia de usabilidade e avaliação de usabilidade, a usabilidade é um dos componentes que integra a aceitabilidade de um sistema, como ilustrado na Figura 2.1.

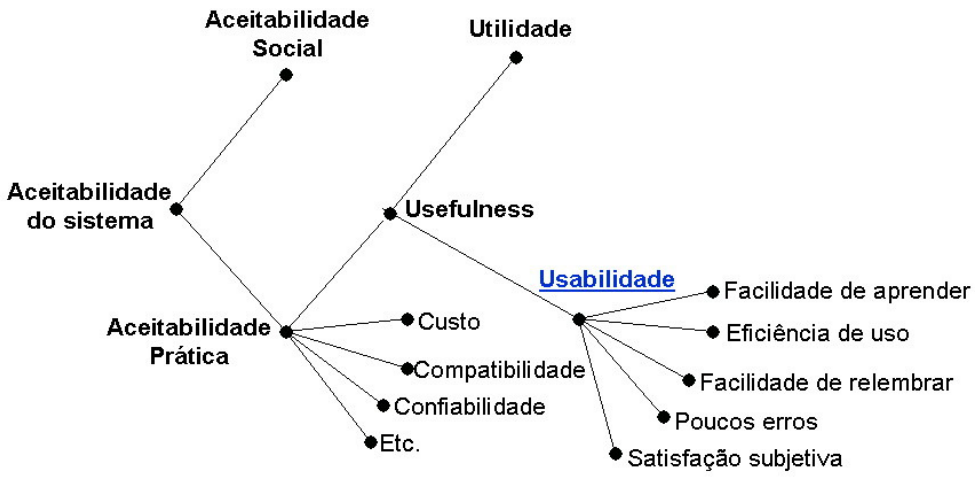


Figura 2. 1. Modelo de aceitabilidade de um sistema.

A usabilidade está relacionada à quão bem os usuários podem utilizar as funcionalidades de um sistema e pode ser decomposta em cinco atributos: facilidade de aprender, eficiência de uso, facilidade de relembrar, poucos erros e satisfação subjetiva do usuário. Estes atributos substituem termos anteriormente adotados que eram imprecisos e difíceis de serem estimados ou mensurados, como os termos "amigável" e "intuitivo".

A **facilidade de aprender** trata do esforço e do tempo necessários para aprender a usar o sistema com determinado nível de competência e desempenho (Prates e Barbosa, 2003). Nielsen (1993) descreve a facilidade de aprender como o mais fundamental atributo de usabilidade, uma vez que a maioria dos sistemas precisa ser de fácil aprendizagem, ainda mais porque a primeira experiência que a maioria das pessoas tem com um sistema novo é exatamente em aprender a usá-lo. O tempo para aprender o conjunto principal de funcionalidades de um sistema e o número de funcionalidades aprendidas podem ser usados como medidas desse atributo de usabilidade.

A **eficiência de uso** é importante, pois, uma vez aprendido o sistema, o usuário torna-se experiente e deve conseguir utilizá-lo de forma eficiente e com alta produtividade. Ou seja, um sistema com eficiência de uso é aquele no qual o usuário é capaz de realizar aquilo que precisa de forma rápida e eficaz. Normalmente, a eficiência de uso pode ser

mensurada analisando o tempo necessário para um usuário completar determinada(s) tarefa(s) no sistema e o número de passos necessários para tal (Prates e Barbosa, 2003).

Além dos usuários novatos e dos experientes, há também os usuários casuais, que já passaram anteriormente pelo processo de aprendizagem, mas não utilizam o sistema tão frequentemente quanto os experientes. Para usuários casuais, portanto, a **facilidade de relembrar** é de grande valia, dispensando a necessidade de aprender tudo outra vez. Notadamente, este atributo de usabilidade é de suma importância para aqueles sistemas usados apenas em períodos específicos, como no caso do programa de declaração anual de imposto de renda. A facilidade de relembrar raramente é avaliada, mas é notável a preocupação das interfaces atuais, nas quais tudo o que for possível é visível – assim, o usuário não precisa lembrar o que está disponível, pois o sistema sempre o relembra quando necessário (Rocha e Baranauskas, 2003). Ainda assim, uma possível forma de medir a facilidade de relembrar em um sistema é através da análise das tarefas completadas com sucesso por um usuário depois de um período específico sem uso (ISO, 1998).

Um sistema com **poucos erros** é aquele que minimiza as ações que levam a um resultado não desejado pelo usuário, o que contribui para que o usuário não perca seu trabalho.

E, finalmente, a **satisfação subjetiva do usuário**, essencial em sistemas de entretenimento e lazer (mas também relevante em ambientes de trabalho) deve ser levada em consideração no design, já que o usuário precisa ficar satisfeito ao usar qualquer tipo de sistema. A satisfação do usuário pode se manifestar através de atitudes positivas ou negativas ou de desconforto durante o uso do sistema (ISO, 1998). Resumidamente, o usuário deve gostar do sistema (Rocha e Baranauskas, 2003). Normalmente, a avaliação deste atributo de usabilidade se dá por intermédio de questionários e pesquisas de opiniões.

Na área de Engenharia de Software, a usabilidade é tida como um requisito de qualidade de sistemas. Pressman (2001) relembra que a norma ISO 9126 foi desenvolvida exatamente para identificar os atributos de qualidade de um software e que a usabilidade é um desses atributos (os outros são: funcionalidade, confiabilidade, eficiência, manutenibilidade e portabilidade). Na ISO 9126, a usabilidade refere-se basicamente à facilidade de uso.

2.2. Métodos de avaliação de usabilidade

Da mesma forma como testes de funcionalidade são comuns para averiguar a robustez da implementação de um sistema, avaliar a usabilidade é necessário para assegurar que o sistema apóia adequadamente os usuários nas suas tarefas (Prates e Barbosa, 2003). E mais: a avaliação de usabilidade deveria ocorrer durante todo o ciclo de vida de um sistema (Rocha e Baranauskas, 2003). Isto significa que se deveria avaliar a usabilidade do sistema desde o início do processo de desenvolvimento para encontrar os problemas de usabilidade e corrigi-los, uma vez que realizar mudanças no sistema vai se tornando cada vez mais difícil e mais caro a medida que o ciclo de desenvolvimento avança (Kennedy, 1989).

Para tal, existem praticamente dois tipos de métodos: os métodos de inspeção (ou métodos analíticos) e os métodos empíricos. Os métodos de inspeção são aqueles nos quais um ou mais avaliadores examinam a interface, julgando-a em busca de problemas de usabilidade, sem a necessidade de verificar a interação de usuários reais com o sistema. Já os métodos empíricos são aqueles nos quais usuários reais participam interagindo com o sistema sendo avaliado enquanto avaliadores realizam a análise de tal interação e dos problemas encontrados.

Apesar da extensa quantidade de métodos existentes para avaliar usabilidade, é muito improvável que um único método seja capaz de encontrar todos os problemas de usabilidade de um sistema. Isto ocorre porque métodos diferentes identificam conjuntos distintos de problemas de usabilidade. Portanto, e para garantir uma boa cobertura de problemas encontrados, recomenda-se avaliar a usabilidade com diferentes métodos, pelo menos um de inspeção e um empírico. Ainda assim, é esperado que haja certo grau de incerteza, mesmo após a execução de múltiplas avaliações com diferentes métodos (Rocha e Baranauskas, 2003).

Mesmo com o grau de incerteza sempre presente, Rocha e Baranauskas (2003) salientam que com a aplicação de métodos de avaliação de usabilidade chega-se a resultados bastante satisfatórios para grande parte dos sistemas.

A seguir, serão descritos alguns dos principais métodos existentes para avaliar usabilidade.

2.2.1. Avaliação Heurística

A avaliação heurística é um dos métodos de inspeção de interfaces mais populares e foi proposto por Nielsen (1994). Trata-se de um conjunto de 10 princípios reconhecidos de usabilidade (que são as heurísticas em si), apresentado na Tabela 2.1.

Tabela 2. 1. Versão revisada das heurísticas de Nielsen.

1. Visibilidade do status do sistema
2. Compatibilidade com o mundo real
3. Controle do usuário e liberdade
4. Consistência e padrões
5. Prevenção de erros
6. Reconhecimento ao invés de relembrança
7. Flexibilidade e eficiência de uso
8. Estética de design minimalista
9. Ajudar os usuários a reconhecer, diagnosticar e corrigir erros
10. Help e documentação

Em uma avaliação heurística, recomenda-se a participação de três a cinco avaliadores, visto que é muito difícil um único avaliador encontrar todos os problemas em uma interface e diferentes avaliadores encontram diferentes problemas de usabilidade. E o custo-benefício não é compensatório quando se utiliza mais do que cinco avaliadores (Nielsen, 1994).

Antes de iniciar a avaliação heurística, pode haver uma fase de preparação para os avaliadores, nos quais se define qual "proposta de design" será avaliada (um protótipo em papel ou um sistema já implementado, por exemplo) (Prates e Barbosa, 2003). E, caso os avaliadores não sejam especialistas no domínio, pode ser necessário um levantamento com tais especialistas para se obter o perfil dos potenciais usuários e cenários de tarefas representativas do sistema durante esta fase de preparação (Prates e Barbosa, 2003).

Após a fase de preparação, inicia-se a avaliação em si, que é divida em duas etapas principais. Na primeira etapa, cada avaliador inspeciona a interface individualmente ao menos duas vezes (Rocha e Baranaukas, 2003), verificando se existem violações das heurísticas, anotando os problemas encontrados e sua localização na interface, gerando uma lista de problemas. Na segunda etapa, há uma sessão de *debriefing* na qual os avaliadores se reúnem a fim de discutir os problemas encontrados e combiná-los em uma única lista final de problemas de usabilidade. Ainda, a cada problema incluído na lista final é atribuído um grau de severidade, que indica a gravidade do problema de usabilidade na interface, de acordo com três critérios: a frequência, o impacto e a persistência do problema na interface (Rocha e Baranauskas, 2003). Atribuir um grau de severidade auxilia os desenvolvedores e designers a priorizar quais problemas devem ser corrigidos primeiro (Prates e Barbosa, 2003). Uma sugestão de graus de severidade tipicamente usados em uma avaliação heurística pode ser vista na Tabela 2.2.

Tabela 2. 2. Graus de severidade.

0. Eu não concordo que isso seja um problema de usabilidade
1. Problema cosmético: não precisa ser corrigido ao menos que haja tempo extra
2. Problema minoritário: deveria ser dada baixa prioridade para sua correção
3. Problema majoritário: importante corrigi-lo, deve ser dada alta prioridade
4. Catástrofe de usabilidade: imperativo corrigi-lo antes do lançamento do produto

As principais vantagens da avaliação heurística são a facilidade em aprendê-la (pode ser aprendida em 4 horas), a rapidez (um dia, no máximo, para a avaliação) e o seu custo – segundo Prates e Barbosa (2003), a avaliação heurística é mais rápida e de menor custo do que a maior parte dos métodos de avaliação de usabilidade. Nielsen (1994) também afirma que a barreira de intimidação, muitas vezes presente em equipes de desenvolvedores, é baixa na avaliação heurística se comparada com outros métodos de avaliação de usabilidade. Além disso, a avaliação heurística pode ser aplicada em qualquer momento do ciclo de vida de um projeto e o entendimento sobre usabilidade dos avaliadores melhora quando compartilham as descobertas com outros avaliadores na sessão de *debriefing* (Nielsen, 1994).

Dentre as desvantagens da avaliação heurística, pode-se citar a fraca contextualização nas condições de uso (Muller et al., 1995), visto que nem todos os avaliadores (especialmente os inexperientes) podem ter uma boa noção de quem serão os reais usuários e suas principais tarefas no sistema, o que pode resultar inclusive em falsos positivos, quando um "problema" apontado na avaliação não é um problema real de usabilidade. Ainda, contam como desvantagens a disponibilidade limitada de avaliadores experientes e a possibilidade muito baixa de se encontrar os aspectos positivos da interface (Law e Hvannberg, 2002). E, como reforça Nielsen (1994), a avaliação heurística não fornece uma maneira sistemática de gerar correções aos problemas de usabilidade encontrados e muito menos garante resultados "perfeitos" ou garante que irá encontrar todos os problemas de usabilidade. Portanto, é sempre prudente aplicar diferentes métodos de avaliação de usabilidade complementares, como, por exemplo, a avaliação heurística e os testes com usuários, que serão abordados futuramente. Nielsen (1994), inclusive, propõe primeiro aplicar a avaliação heurística para se encontrar aqueles problemas mais "óbvios", corrigi-los e então aplicar testes com usuários para descobrir os problemas que ainda persistem.

2.2.2. Percurso Cognitivo

Outro método de inspeção de usabilidade é o percurso cognitivo. Diferente da avaliação heurística, orientada ao produto (Muller et al., 1995), o percurso cognitivo é orientado à tarefa: o analista, como é chamado o avaliador no percurso cognitivo, explora os elementos da interface simulando passo-a-passo o comportamento do usuário final para realizar uma dada tarefa.

Conforme mencionado anteriormente, Nielsen (1993) destaca a facilidade de aprender como o atributo mais fundamental de usabilidade. Portanto, ao explorar a interface de um novo sistema, seria interessante que um usuário fosse capaz de aprendê-la e, de fato, muitas pessoas preferem aprender as funcionalidades de um sistema enquanto trabalham em suas tarefas típicas (Prates e Barbosa, 2003), adquirindo conhecimento sobre as funcionalidades do sistema de acordo com a necessidade, ao invés de consultarem manuais ou participarem de treinamentos (Rocha e Baranauskas, 2003). Assim, o foco do percurso cognitivo é exatamente encontrar problemas que os usuários terão ao usar pela primeira vez a interface (Lewis e Rieman, 1994), ou seja, enquanto estão aprendendo a usá-la.

Semelhante à avaliação heurística, o percurso cognitivo também é dividido em duas fases. Na primeira fase, de preparação, o analista define quem são os usuários do sistema (perfil dos usuários), quais tarefas devem ser analisadas, qual é a sequência correta de ações necessária para se realizar cada tarefa e como a interface está definida (descrição textual, protótipo, etc.). Na segunda fase, de análise, o analista examina cada uma das ações necessárias para cumprir cada uma das tarefas e tenta contar uma história verossímil de como o usuário escolheria tal ação (Rocha e Baranauskas, 2003). Para isto, o analista deve responder quatro questões:

Q1. Os usuários farão a ação correta para atingir o resultado desejado?

Q2. Os usuários perceberão que a ação correta está disponível?

Q3. Os usuários associarão a ação correta com o resultado desejado?

Q4. Se a ação correta for executada, os usuários perceberão que foi feito um progresso em relação à tarefa desejada?

Ao tentar responder estas perguntas, e baseado no perfil dos usuários, o analista é capaz de identificar conflitos na forma como designers e usuários entendem como uma tarefa deve ser realizada, palavras inadequadas na interface (menus, rótulos de botões, etc.) e *feedback* insatisfatório (Rocha e Baranauskas, 2003).

Apesar de ajudar a identificar problemas relacionados com a facilidade de aprender, o percurso cognitivo pode ser demorado e muito detalhista, e, portanto, é adotado para avaliar apenas pequenos cenários críticos do sistema (Rocha e Baranauskas, 2003). Ainda, por focar apenas um atributo de usabilidade, a avaliação dos demais atributos pode não ser satisfatória, como a consistência global da interface, reforçando novamente a necessidade de adotar diferentes métodos para avaliar a usabilidade (Rocha e Baranauskas, 2003).

2.2.3. Testes com usuários

Outro método muito tradicional de avaliação de usabilidade, tido como fundamental e que se recomenda ser aplicado de forma a complementar aos métodos de inspeção são os testes com usuários, ou testes de usabilidade.

Trata-se de um método empírico, que envolve um grupo de potenciais usuários do sistema. Comparando com a avaliação heurística, os testes com usuários não são tão rápidos e nem tão baratos. Todavia, testes de usabilidade tendem a produzir resultados mais precisos e objetivos do que a avaliação heurística (Law e Hvannberg, 2002). Ainda, de acordo com Muller et al. (1995), testes são orientados a tarefas e a avaliação heurística orientada ao produto, o que justifica a adoção de testes para complementar os resultados da avaliação heurística.

Os testes com usuários podem ser realizados em ambientes controlados, como um laboratório, ou no ambiente de trabalho dos usuários. Os testes em ambientes controlados têm como principal vantagem o isolamento de fatores externos ao ambiente, como por exemplo, potenciais interrupções do usuário enquanto o mesmo está executando uma tarefa (Prates e Barbosa, 2003). Entretanto, estando em um ambiente controlado e artificial, pouco parecido com o seu ambiente de trabalho, os usuários podem não se sentir confortáveis e

nem todas as variáveis relevantes relacionadas às tarefas dos usuários podem ter sido levantadas, uma vez que se está fora do contexto real de uso (Prates e Barbosa, 2003).

Os testes no próprio ambiente de trabalho dos usuários, também conhecidos como pesquisa de campo, visam exatamente superar esta desvantagem dos testes em ambientes controlados, uma vez que se procura avaliar como a tecnologia é usada em um ambiente real (Prates e Barbosa, 2003). Neste caso, abordagens etnográficas de observação e entrevistas podem ser adotadas para se extrair dados relevantes a respeito do uso do sistema.

Mesmo que não haja um laboratório para a realização de testes com usuários em ambientes controlados, Rocha e Baranauskas (2003) relembram que isto não deve ser uma condição impeditiva à aplicação dos mesmos. Desde que devidamente preparados, quase todas as técnicas e processos usados em testes com usuários podem ser realizados nos mais diversos locais (Rocha e Baranauskas, 2003).

Na preparação para os testes, é preciso especificar o objetivo da avaliação, ou seja, quais atributos de usabilidade serão analisados, por exemplo, a eficiência de uso ou a satisfação do usuário. Adicionalmente, é necessário especificar o que deve ser avaliado, mais precisamente, quais os pontos críticos, quais tarefas básicas ou muito frequentes de uso do sistema, sempre tentando considerar as tarefas representativas, mais próximas da realidade de uso do sistema (Prates e Barbosa, 2003).

Quanto às tarefas, Prates e Barbosa (2003) recomendam ainda que as primeiras tarefas a serem executadas pelos participantes sejam simples e fáceis para deixá-los mais confiantes. Da mesma forma, as últimas tarefas também deveriam ser simples e fáceis para que o participante termine os testes sentindo-se bem com o seu desempenho.

Além da definição de objetivos e tarefas, durante a preparação dos testes, é preciso recrutar participantes para os testes. Os participantes selecionados para os testes devem ter o perfil mais próximo possível dos usuários reais previstos para o sistema. Para Lewis e Rieman (1994), os melhores participantes de testes são aqueles que se espera que sejam usuários reais do sistema.

O número de participantes sugerido por Nielsen (2000) é de 3 a 5, com vista a obter o melhor custo-benefício. Com 5 participantes é possível encontrar aproximadamente 85% dos problemas de usabilidade de um sistema, e ao incluir mais usuários o ganho no número

de novos problemas encontrados é pouco significativo – para encontrar todos os problemas de usabilidade, teoricamente, são necessários 15 participantes (Nielsen, 2000). A curva ilustrada na Figura 2.2, de Nielsen (2000), mostra a relação entre o número de participantes e a quantidade de problemas de usabilidade encontrados.

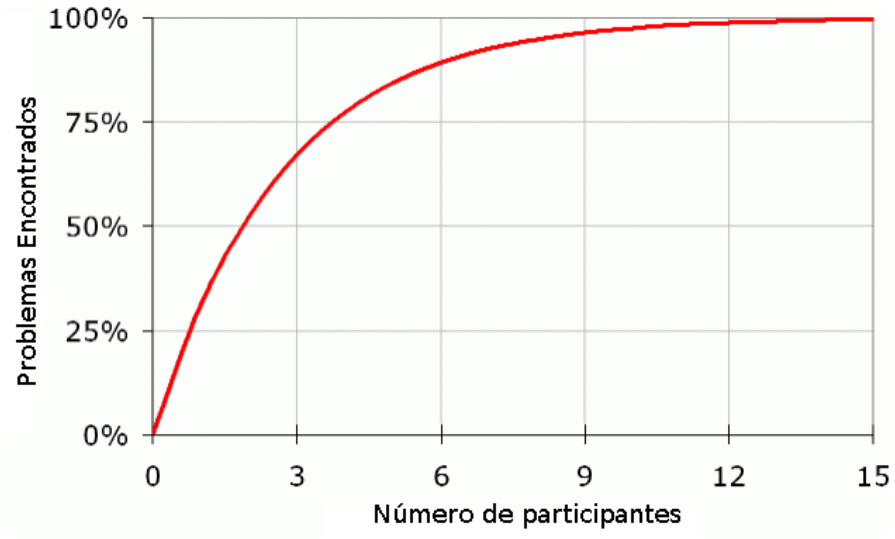


Figura 2. 2. Relação entre número de participantes e a quantidade de problemas de usabilidade encontrados em testes com usuários (Nielsen, 2000).

Tipicamente, os participantes de testes de usabilidade são voluntários. Assim, e antes de iniciar os testes, é de praxe que cada participante concorde formalmente com um termo de consentimento, que declara os objetivos dos testes (de avaliar a interface, não o usuário), a garantia de anonimato do participante e o direito do mesmo em interromper o teste a qualquer momento que desejar. Adicionalmente, em algumas organizações e em alguns países, para se realizar testes com usuários, pode ser necessária aprovação prévia por comitês de ética, responsáveis por lidar com pesquisas que envolvem seres humanos (Lewis e Rieman, 1994) – no Brasil, é comum, mas ainda não obrigatório, a adoção dos mesmos procedimentos legais aplicados em experimentos de outras áreas que demandam pela participação de seres humanos, especialmente a medicina. Questões de privacidade dos dados obtidos também podem e devem ser consideradas – por exemplo, Lewis e Rieman afirmam que, quando realizam testes com usuários, evitam gravar em vídeo os rostos dos

participantes e não colocam os nomes dos mesmos junto aos dados coletados, preferindo identificá-los através de números.

Após a etapa de preparação de testes, segue a execução dos testes em si, que se dá de forma individual por participante. Um documento impresso contendo o conjunto de tarefas representativas selecionadas pelos avaliadores é entregue ao participante, que deve tentar executá-las no sistema dentro de um período de tempo pré-determinado pelos avaliadores. Enquanto o participante percorre as tarefas no sistema, os avaliadores observam e realizam anotações. Também, recursos para gravação de áudio, vídeo e registro de uso do sistema (*logs*) podem ser adotados para análise posterior ou para confirmar alguma anotação imprecisa por parte dos avaliadores.

Durante a realização dos testes, recomenda-se que o avaliador não auxilie o usuário com as tarefas. Segundo Prates e Barbosa (2003), o ideal é que perguntas sobre o sistema e a execução das tarefas não sejam respondidas, que o avaliador seja mais um observador para que não distorça os resultados dos testes (Lewis e Rieman, 1994). A ajuda ao participante deveria ser oferecida apenas em casos em que não há como se prosseguir com os testes (Lewis e Rieman, 1994).

Apesar de não ser recomendado auxiliar o participante a executar uma tarefa, o avaliador pode estimulá-lo a pensar em voz alta (*think aloud*), ou seja, verbalizar tudo aquilo que está pensando a medida que tenta cumprir com a tarefa (Prates e Barbosa, 2003). A técnica do pensar em voz alta permite ao avaliador entender o modelo mental do usuário e descobrir os problemas na interface. Porém, alguns pesquisadores acreditam ser uma técnica pouco natural, inibidora do participante, e sugerem como alternativa o método de co-descoberta, ou *co-discovery learning* (Kennedy, 1989), no qual dois participantes juntos tentam realizar as tarefas e conversam entre si para tal.

A técnica do pensar em voz alta resulta em uma lista com as dificuldades que foram encontradas pelos participantes (Lewis e Rieman, 1994). Porém, medidas quantitativas de desempenho também podem ser observadas e analisadas pelos avaliadores durante os testes, tais como: o tempo necessário para terminar uma tarefa, o número de erros encarados pelo participante, a porcentagem de participantes que conseguiram se recuperar de um erro, o número de vezes em que foi requisitada ajuda, seja do avaliador, seja do *help* do sistema (Rocha e Baranauskas, 2003; Prates e Barbosa, 2003).

Lewis e Rieman (1994) afirmam que adotar tanto a técnica do pensar em voz alta quanto a de medição quantitativa de desempenho pode ocasionar alguma distorção nos resultados encontrados, uma vez que o pensar em voz alta potencialmente afeta o desempenho dos participantes, especialmente o tempo necessário para cumprir uma tarefa e o número de erros, pois os participantes tendem a demorar mais nas tarefas por terem que verbalizar seus pensamentos e, como pensam mais em como fazer as coisas, acabam por cometer menos erros.

Em geral, as medidas quantitativas de desempenho e os comentários e dúvidas do participante são anotados pelos avaliadores enquanto ele realiza as tarefas. Além disso, e como mencionado anteriormente, em algumas oportunidades, gravações em vídeo e outros meios de registro de uso do sistema (como o *log* do sistema, por exemplo) podem ser adotados para tentar tornar o processo de observação mais discreto ou para confirmar alguma imprecisão nas anotações, mas o tempo de análise deste tipo de dado é considerado bastante elevado (Holzinger, 2005).

Finalmente, ao término dos testes, os avaliadores podem convidar os usuários a expor suas opiniões, seja na forma de uma conversa informal ou entrevista, seja respondendo um questionário. Assim, pode-se realizar um levantamento sobre a satisfação subjetiva do usuário com o sistema.

Capítulo 3

Acessibilidade e Design Universal

Serão abordados neste capítulo alguns aspectos do design universal e da acessibilidade, como seus princípios, sua origem e sua importância no mundo atual.

3.1. Pessoas com Deficiências

Antes de falar a respeito de acessibilidade e de design universal, é preciso conhecer quem são as pessoas por trás destes conceitos, suas necessidades, especificidades e habilidades. Ou seja, quem são os potenciais usuários alvo de um design acessível ou universal.

A Organização Mundial de Saúde (OMS, 2009) classifica detalhadamente as deficiências existentes na *International Classification of Functioning, Disability and Health* (ICFDH). Também, define deficiências como problemas nos sentidos (audição e visão, principalmente) ou na estrutura do corpo, como um desvio ou uma perda, e destaca que as deficiências podem ser permanentes ou temporárias, progressivas, regressivas ou estáveis, intermitentes ou contínuas, e apresentar diversos níveis de comprometimento (severa, moderada, leve, etc.).

De forma mais simplificada, a legislação brasileira classifica as deficiências entre física, auditiva, visual, mental e múltipla (IDT, 2009). A deficiência física agrupa as diversas formas de paralisia, como a paraplegia (perda dos movimentos dos membros inferiores), a tetraplegia (perda dos movimentos dos membros inferiores e superiores) e a hemiplegia (paralisia de um dos lados do corpo, muitas vezes como sequela de um acidente

vascular cerebral), bem como amputações, ausências de membros e a paralisia cerebral, caracterizada por dificuldades motoras. Já a deficiência auditiva é a perda, parcial ou total, da audição, enquanto a deficiência visual é caracterizada pela cegueira e pela visão subnormal ou baixa visão, respectivamente a ausência total ou quase total de visão e a perda severa da mesma (IES, 2009). A deficiência mental é caracterizada por dificuldades de aprendizagem e pelo baixo nível intelectual que, dependendo do grau de comprometimento, também podem ocasionar dificuldades no desenvolvimento e na interação social (Santana, 2007). Porém, alguns autores preferem a denominação "deficiências cognitivas" para classificar os transtornos e as síndromes que envolvem dificuldades ou déficits de aprendizagem, atenção, memória e leitura, em que se incluem o autismo, o transtorno de déficit de atenção, a dislexia, a Síndrome de Down, entre outros (WebAIM, 2009). Finalmente, a deficiência múltipla é a associação de duas ou mais deficiências.

Um fato a se destacar é que muitas deficiências podem ser congênitas ou podem ser resultado de doenças ou de acidentes. Assim, uma pessoa que nasceu com uma determinada deficiência pode apresentar habilidades, necessidades e preferências diferentes de outra pessoa que adquiriu a mesma deficiência quando já adulta. Por exemplo, considerando dois indivíduos fictícios, João e Maria, ambos com 20 anos de idade. João nasceu surdo e Maria ficou surda aos 19 anos. João, por sempre ter sido surdo, desde a sua infância comunica-se pela língua de sinais e, de fato, a tem como a sua primeira língua, sendo a língua escrita e falada (ou seja, o Português) sua segunda língua. Assim, João prefere comunicar-se pela língua de sinais e sente alguma dificuldade em ler e escrever em Português. Já Maria, pelo seu contato prévio com a língua escrita e falada antes de ficar surda, tem mais facilidade em ler e escrever em Português, mas tem um domínio muito restrito da língua de sinais. Ou seja, ao contrário de João, Maria não sente dificuldades em ler e escrever em Português, mas não consegue comunicar-se muito bem pela língua de sinais, pois começou a aprendê-la e a usá-la há menos tempo que João.

Obviamente, o nível de comprometimento da deficiência também é outro fator que leva as pessoas com o mesmo tipo de deficiência a terem diferentes habilidades, necessidades e preferências. Por exemplo, novamente considerando indivíduos fictícios com deficiência visual, Paula, Tadeu e José. Paula e Tadeu são cegos enquanto José tem

visão subnormal. Paula não apresenta qualquer resíduo visual (ou seja, é completamente cega) desde que nasceu e por isto prefere ler um livro em Braille. Por outro lado, José, como mantém um resíduo visual, pode ler um livro escrito em Português com o auxílio de uma lupa especial prescrita pelo seu oftalmologista. Já Tadeu, que perdeu completamente sua visão em um acidente recente, ao invés de ler livros impressos em Português ou em Braille, prefere ouvir a versão em áudio sempre que possível e disponível, uma vez que ainda não aprendeu o Braille.

Enfim, é importante notar que cada deficiência tem suas especificidades e, dentro de cada tipo de deficiência, os indivíduos são diferentes e apresentam necessidades e habilidades distintas. Ou seja, é impossível definir um perfil único para representar o conjunto de pessoas com deficiências.

3.2. Origens históricas da acessibilidade e do design universal

Historicamente, pessoas consideradas "não normais" são estigmatizadas e marginalizadas pela sociedade. Tal comportamento, inclusive, desencadeou ações cruéis e violentas no passado.

Em Esparta, na Grécia Antiga, os "monstros e deformados" eram jogados do alto do monte Taígeto. Celtas permitiam que filhos matassem os pais quando estes estivessem velhos e doentes. Nas Américas, certos povos indígenas tinham o hábito de abandonar os incuráveis e sacrificar recém-nascidos "desgraçados" (Wendt, 2001). E, durante a Idade Média, a imagem da pessoa com deficiência representava o castigo, a imperfeição, o possuído pelo demônio, justificando a segregação e a rejeição da sociedade (Vizim, 2003).

Na arquitetura e no design, tal marginalização também se fez presente. O design de praticamente tudo o que existia no mundo era exclusivamente feito por e para pessoas consideradas dentro da "normalidade estática": pessoas que nunca envelhecem, nunca engordam, nunca ficam cansadas e nunca precisam mudar sua rotina (Covington e Hannah, 1997). Como resultado, ainda hoje é comum encontrar prédios públicos sem acesso

adequado para usuários de cadeiras de rodas (ausência de rampas e/ou elevadores) e bibliotecas sem um único livro em Braille, por exemplo.

Aos poucos, porém, o avanço da ciência contribuiu para a alteração da imagem da pessoa com deficiência. Tanto que, em meados do século XX, surgiram os ideais de design universal e de acessibilidade. Tal sensibilização do design em geral, particularmente da arquitetura, foi impulsionada principalmente por alterações demográficas.

Os progressos da medicina e as melhorias nas condições sanitárias nas cidades proporcionaram ao ser humano melhor qualidade de vida, maior resistência a doenças antes incuráveis e, consequentemente, maior longevidade. Tanto é assim que há um notório aumento da população idosa em várias partes do mundo, mesmo em países em desenvolvimento como o Brasil. Entretanto, o envelhecimento da população vem frequentemente acompanhado de uma inevitável redução nas capacidades sensoriais e físico-motoras. Além do aumento natural da população idosa com potenciais dificuldades motoras e sensoriais, as guerras do século XX resultaram em milhares de veteranos amputados, tetraplégicos, paraplégicos, surdos, cegos, etc.. Ainda, alterações na vida cotidiana também contribuíram para aumentar o número de pessoas fora da "normalidade estática". Como relata Goldsmith (2001), hoje em dia há muitos casos de pessoas que adquiriram deficiências físicas após sofrerem acidentes de trânsito ou praticarem determinadas atividades esportivas.

Com o aumento da população de pessoas com deficiências, ficaram cada vez mais evidentes as barreiras de acesso existentes em espaços físicos, produtos e serviços. E uma das reações da sociedade se deu através dos movimentos civis para a adoção de medidas para a adaptação dos meios físicos em meados do século XX.

Tim Nugent, responsável pelo programa de reabilitação da *University of Illinois* do campus de Champaign-Urbana nos anos 1950s, acreditava que a América era uma terra de oportunidades e todos aqueles que lutam e trabalham são recompensados e obtêm sucesso. Então, a pessoa com deficiência também deveria lutar e trabalhar para seu próprio sucesso, não dependendo da caridade e da boa vontade dos outros. Para isto, não deveriam existir barreiras para o seu sucesso (Goldsmith, 2001).

Com este ideal e muita insistência, Nugent conseguiu autorização para adequação de todos os 200 prédios do campus para se tornarem acessíveis a alunos cadeirantes,

possibilitando que estes alunos pudessem assistir às mesmas aulas que os demais. E, para realizar todas as adequações, Nugent acabou criando padrões de acessibilidade para a arquitetura. Mais tarde, o trabalho iniciado por Nugent acabou se tornando o primeiro conjunto de padrões de arquitetura em nível nacional, conhecido como *American Standard Specifications for Making Buildings and Facilities Accessible to, and Usable by, the Physically Handicapped*, que influenciou leis e normas no resto do mundo.

Ainda na segunda metade do século XX, nos Estados Unidos, o movimento dos direitos das pessoas com deficiências (*Disability Rights Movement*) teve como resultado leis que proíbem qualquer forma de discriminação e a garantia de acesso à educação, transporte e comunicações. Estes esforços contribuíram para uma evolução dos padrões de acessibilidade e a criação de novas normas para a construção de prédios "livres de barreiras".

Colocadas em prática, em pouco tempo percebeu-se que algumas das normas para o design livre de barreiras poderiam beneficiar não apenas as pessoas com deficiência como também o restante da população. Uma guia rebaixada na calçada para facilitar a locomoção de um cadeirante também auxilia uma mãe empurrando um carrinho de bebê e uma pessoa idosa que necessita de um andador para se apoiar, por exemplo. E assim surgiu a expressão design universal.

O termo design universal foi usado pela primeira vez pelo arquiteto e designer Ronald Mace. Segundo Mace (1998), o design universal é o desenvolvimento de produtos e ambientes agradáveis esteticamente e usáveis por todas as pessoas, o máximo possível, independente de idade, habilidade ou status social, sem a necessidade de adaptação ou design especial ou exclusivo. E, para Mace, universal e livre de barreiras são coisas diferentes, já que um projeto livre de barreiras é dirigido especificamente a pessoas com deficiências, mas não a todas as pessoas (Mace, 1998). Como exemplo, Mace cita que um produto deveria ser usável tanto por canhotos quanto por destros se desenvolvido de acordo com os ideais de design universal, mas que isto seria irrelevante se fossem considerados apenas os princípios de design livre de barreiras.

Outros especialistas têm a mesma opinião de Mace sobre o design universal. "O papel do design universal não é criar produtos especiais, uma vez que o design universal não é exclusivo e sim inclusivo, é a idéia de que todo mundo deveria poder ter acesso a

tudo a qualquer momento; tarefa difícil, mas não impossível" (Covington e Hannah 1997, pág. 14).

3.3. Princípios do Design Universal

Mace juntou-se a uma equipe constituída por arquitetos, engenheiros e designers de produtos e ambientes e estabeleceu os sete princípios do design universal. Tais princípios podem ser aplicados para avaliar um design já existente, guiar processos de design e educar tanto designers quanto usuários sobre as características de produtos e ambientes mais usáveis (Connel et al., 1997).

Cada princípio do design universal possui uma lista de elementos chaves, ou *guidelines*, que devem estar presentes no design para satisfazer o referido princípio, com a ressalva dos autores de que nem todas as *guidelines* podem ser relevantes para todo o tipo de design. Os sete princípios são descritos a seguir, conforme publicado por Connel et al. (1997).

Princípio um: Equidade de uso

O design é útil e adequado para pessoas com diversas habilidades.

Guidelines:

1.a Oferecer as mesmas maneiras de uso para todos os usuários: idêntica sempre que possível; equivalente caso contrário.

1.b Evitar segregar ou estigmatizar quaisquer usuários.

1.c Providências a respeito de privacidade e segurança deveriam ser iguais para todos os usuários.

1.d Fazer o design atraente para todos os usuários.

Princípio dois: Flexibilidade no Uso

O design acomoda uma ampla faixa de habilidades e preferências individuais.

Guidelines:

2.a Oferecer escolhas de maneiras de utilização.

2.b Acomodar o acesso e o uso por destros e canhotos.

2.c Facilitar a exatidão e a precisão do usuário.

2.d Oferecer adaptabilidade aos tempos de ação e de resposta do usuário.

Princípio três: Simples e Intuitivo

O uso do design é de fácil entendimento, independente da experiência, do conhecimento, da linguagem e do atual nível de concentração do usuário.

Guidelines:

3.a Eliminar complexidade desnecessária.

3.b Ser consistente com as expectativas e a intuição do usuário.

3.c Acomodar uma ampla faixa de nível de alfabetização e habilidades com a língua.

3.d Organizar informação consistente com sua importância.

3.e Oferecer *prompt* e *feedback* efetivos durante e depois de uma tarefa ser finalizada.

Princípio quatro: Informação Perceptível

O design comunica efetivamente informação necessária ao usuário, independente das condições do ambiente ou das habilidades sensoriais do usuário.

Guidelines:

4.a Usar diferentes modos (pictório, verbal, tátil) para apresentação redundante de informação essencial.

4.b Oferecer contraste adequado entre informação essencial e o restante ao seu redor.

4.c Maximizar a legibilidade de informação essencial.

4.d Diferenciar elementos de forma que possam ser descritos (ou seja, tornar fácil para dar instruções ou direções).

4.e Oferecer compatibilidade com uma ampla variedade de técnicas ou dispositivos usados por pessoas com limitações sensoriais.

Princípio cinco: Tolerância a Erros

O design minimiza danos e consequências adversas de ações acidentais ou não intencionadas.

Guidelines:

5.a Organizar os elementos de forma a minimizar perigos e erros: elementos mais usados ficam mais acessíveis; elementos perigosos são eliminados, isolados ou protegidos.

5.b Oferecer avisos de perigos e erros.

5.c Prover funcionalidades tolerantes a falhas.

5.d Desencorajar ação inconsciente em tarefas que exigem monitoramento.

Princípio seis: Baixo Esforço Físico

O design pode ser usado eficiente e confortavelmente com um mínimo de fadiga.

Guidelines:

6.a Permitir ao usuário manter uma posição corporal neutra.

6.b Usar força não excessiva para operação.

6.c Minimizar ações repetitivas.

6.d Minimizar esforço físico.

Princípio sete: Tamanho e Espaço para Aproximação e Uso

Tamanho e espaço apropriado é disponibilizado para aproximação, alcance, manipulação e uso não importando o tamanho do corpo do usuário, sua postura ou mobilidade.

Guidelines:

7.a Oferecer uma clara linha de visão para elementos importantes para quaisquer usuários sentados ou em pé.

7.b Tornar alcançáveis confortavelmente todos os componentes para quaisquer usuários sentados ou em pé.

7.c Acomodar variações para diferentes tamanhos de mão.

7.d Disponibilizar espaço adequado para o uso de tecnologias assistivas[1] ou assistência pessoal.

3.4. Acessibilidade em sistemas de informação

Acessibilidade é um pouco difícil de definir ou, pelo menos, de se escolher uma dentre tantas definições existentes, visto que há, atualmente, dezenas de definições para acessibilidade em sistemas de informação, algumas mais próximas do design livre de barreiras e outras mais próximas do design universal.

Uma definição claramente inspirada no design livre de barreiras é a de Bergman e Johnson (1995), que afirmam que oferecer acessibilidade significa remover barreiras que impedem pessoas com deficiências de participar de atividades substanciais, incluindo o uso de serviços, produtos e informação.

Em se tratando de acessibilidade para a Web, Henry (2005) define acessibilidade como permitir que pessoas com deficiência possam usar efetivamente a Web. Ou seja, permitir que pessoas com deficiência possam perceber, entender, navegar, interagir e contribuir com a Web (Henry, 2005). Apesar de apenas citar pessoas com deficiências em sua definição de acessibilidade para a Web, Henry ressalta que acessibilidade para a Web

[1] Tecnologias assistivas serão discutidas posteriormente neste capítulo no subtópico 3.3.1.

beneficia também outras pessoas, incluindo idosos com alterações em suas habilidades por conta do envelhecimento (Henry, 2005). De forma similar, Leporini e Paternò (2004) afirmam que um site pode ser dito acessível se puder ser usado por todos, inclusive por pessoas com deficiências.

Uma definição mais ampla de acessibilidade para sistemas de informação, compatível com os ideais de design universal, foi proposta pelo grupo de especialistas do Usability Junction (2002), para o qual acessibilidade é o desenvolvimento de sistemas de informação flexíveis o suficiente para acomodar as necessidades dos usuários, independente de idade, deficiência ou tecnologia utilizada, promovendo uma experiência igual para todos. Por tecnologia utilizada, pode-se entender apenas aquelas destinadas às pessoas com deficiências, sendo conhecidas por tecnologias assistivas ou ajudas técnicas, definidas por Hogetop e Santarosa (2001) como o conjunto de recursos que, de alguma maneira, contribuem para proporcionar às pessoas com deficiências maior independência, qualidade de vida ou inclusão social, potencializando suas capacidades. Entretanto, não se pode esquecer de que celulares, PDAs, dentre outros, também podem ter acesso às mesmas informações.

Obrenovic et al. (2007) relembram também que há diversas variações para se falar praticamente da mesma coisa: acessibilidade universal, design inclusivo, design para todos... Para os referidos autores, todas essas variantes objetivam produzir sistemas que possam ser usados por **todas as pessoas**, independente de habilidades físicas e cognitivas.

Partindo da premissa de que a acessibilidade em sistemas de informação tem de alguma forma inspiração nos ideais de design universal de Mace, pode ser possível afirmar que, ao melhorar a acessibilidade de um sistema, todos serão beneficiados, não apenas pessoas com deficiências. E, sendo assim, deve existir uma relação entre acessibilidade e usabilidade, conforme indicam pesquisas como as de Obrenovic et al. (2007) e nossas (Tanaka et al., 2005; Tanaka et al., 2006).

Apesar da iminente relação entre acessibilidade e usabilidade, não há consenso sobre a exata relação entre os dois conceitos. Para Leporini e Paternò (2004), usabilidade e acessibilidade são conceitos intimamente relacionados, porém acessibilidade é disponibilizar um sistema para uma faixa mais ampla da população enquanto a usabilidade está mais centrada em tornar a experiência do usuário mais eficiente e mais agradável.

Ainda, afirmam que a acessibilidade técnica, que assegura o funcionamento do sistema com tecnologias assistivas, é uma pré-condição para a usabilidade (Leporini e Paternò, 2004).

Simplificar e assumir que acessibilidade e usabilidade são conceitos equivalentes também é fortemente contestado por Powlik e Karshmer (2002). De acordo com os autores, tal hipótese é similar a assumir que *broadcasting* equivale a uma comunicação efetiva.

Independente da relação entre acessibilidade e usabilidade, a acessibilidade é um atributo de qualidade que não pode mais ser ignorado pelos sistemas de informação, não importam quais sejam. E, sendo assim, Theofanos e Redish (2003) levantaram cinco razões para justificar um design acessível para a Web, que pode ser estendido para qualquer sistema de informação.

Primeiro, e diferente do que muitos imaginam, as deficiências não afetam um pequeno e irrelevante grupo de pessoas. Pelos dados de Theofanos e Redish (2003), 750 milhões de pessoas no mundo apresentam algum tipo de deficiência. No Brasil, o censo populacional de 2000 constatou que, do total de 169.799.170 de brasileiros, 24.537.984 possuem algum tipo de deficiência (IBGE, 2000), o que representa 14,6 % da população. A Tabela 3.1 resume os dados do censo a respeito da população brasileira com alguma deficiência e a porcentagem da população brasileira que apresenta estas deficiências. Levando em consideração outros dados do censo populacional de 2000, o número de brasileiros com deficiências equivale a mais de 24 vezes o número de habitantes na cidade de Campinas/SP, é quase do mesmo tamanho de toda a população da região Sul do Brasil e aproximadamente 1,5 vezes maior do que a população da Holanda, que era de cerca de 16 milhões de habitantes em 2000.

Tabela 3. 1. População brasileira com deficiência (IBGE, 2000)[2].

Tipo de Deficiência	Total	Porcentagem da população
Mental	2.848.684	1,678
Física	1.422.224	0,838
Tetraplegia, paraplegia ou hemiplegia permanente	955.287	0,563
Falta de membro ou de parte dele	466.936	0,275
Motora	7.879.601	4,641
Incapaz de caminhar ou subir escada	588.201	0,346
Grande dificuldade permanente de caminhar ou subir escada	1.799.917	1,060
Alguma dificuldade permanente de caminhar ou subir escada	5.491.482	3,234
Auditiva	5.750.809	3,387
Incapaz de ouvir	176.067	0,104
Grande dificuldade permanente de ouvir	860.889	0,507
Alguma dificuldade permanente de ouvir	4.713.854	2,776
Visual	16.573.937	9,761
Incapaz de enxergar	159.824	0,094
Grande dificuldade permanente de enxergar	2.398.472	1,413
Alguma dificuldade permanente de enxergar	14.015.641	8,254

Segundo, e novamente contrariando o senso comum, Theofanos e Redish afirmam que acessibilidade pode, sim, ser um ótimo negócio. Não apenas por conta do tamanho da população, mas também por conta de sua renda estimada. Nos Estados Unidos, os rendimentos anuais das pessoas com deficiência chegam a US$ 175 bilhões (Theofanos e Redish, 2003). Além disso, há casos em que a falta de um nível adequado de acessibilidade levou a processos judiciais que, invariavelmente, acarretam prejuízos tanto financeiros quanto na imagem das empresas, como no caso da "Target.com" nos Estados Unidos, que se viu envolvida em um processo judicial por ter seu site inacessível a pessoas com deficiências visuais (DRA, 2006).

Terceiro, como já destacado anteriormente, o número de pessoas com deficiências e dificuldades físico-motoras e sensoriais tende a aumentar, visto que a expectativa de vida

[2] É possível observar que a soma dos dados na tabela ultrapassa o total de pessoas com deficiências de acordo com os dados do IBGE, uma vez que pessoas com múltiplas deficiências aparecem em duas ou mais linhas. Ainda, a tabela exibe a classificação original do IBGE, apesar de muitas vezes ser um tanto quanto superficial ou generalista – por exemplo, não há informação a respeito do que se considerou deficiência mental no censo populacional. Dados mais precisos e recentes ainda não estavam disponíveis até a escrita desta tese de doutorado.

das pessoas está aumentando e, naturalmente, com o envelhecimento muitas vezes ocorrem perdas motoras e sensoriais significativas (Theofanos e Redish, 2003).

Quarto, pessoas com deficiência gostam de tecnologias e de navegar pela Web como quaisquer outras pessoas. Elas se sentem livres, independentes e autônomas quando conseguem acessá-las plenamente (Theofanos e Redish, 2003). E mais: de acordo com Bailey e Burd (2006), oferecer serviços online para pessoas com deficiências pode ser de enorme valia, principalmente nos casos em que há dificuldades no acesso físico.

E, finalmente, o quinto e principal argumento levantado por Theofanos e Redish (2003): moralmente, é a coisa certa a ser feita!

Naturalmente, a visão de que uma interface deveria ser acessível para todos pode parecer utópica. E, de fato, há questionamentos neste sentido com os quais há de se concordar. Hull (2004), afirma que um site, para ser acessível para todos, deveria estar disponível em todas as línguas e dialetos do mundo – existem mais de seis mil línguas no mundo e nem todas possuem forma escrita! Acessível para todos, literalmente, é muito custoso e próximo ao impossível (Hull, 2004).

Outro aspecto relevante e que merece atenção é que, quando se desenvolve um sistema para um determinado público com deficiência, ele não necessariamente será acessível para todos. Por exemplo, um site totalmente textual pode ser facilmente lido por um leitor de telas, sendo acessível para cegos, mas não se pode dizer o mesmo para pessoas surdas que não dominam a língua escrita e pessoas com deficiências cognitivas, para as quais o recomendado é exatamente privilegiar a comunicação não-verbal/não-textual (Lewis, 2006). Powlik e Karshmer (2002) também concordam que texto é geralmente mais acessível, exceto para pessoas com dificuldades de aprendizagem e déficit de atenção, que necessitam de um esforço maior para ler e memorizar.

A grande falha do exemplo anterior está justamente no fato do sistema ser específico para um público. Sendo **específico** ou **exclusivo**, não há como ser **universal** ou **inclusivo**. E, além disso, notam-se conflitos: para cegos, o ideal é oferecer uma interface toda textual, acessível via leitor de telas, enquanto para pessoas com deficiências cognitivas, com dificuldades de leitura e memorização ou surdas sem o domínio da língua escrita, a interface deveria privilegiar elementos não textuais!

Na prática, portanto, pode-se dizer que Hull (2004) tem razão ao afirmar que o design universal de sistemas de informação é, de fato, muito próximo à utopia. Quando se fala que um sistema é acessível ou inclusivo, é importante saber para quem ele é acessível ou quem ele está incluindo – que, certamente, não serão todos!

Entretanto, vale ressaltar que, desde a sua concepção por Mace, o design universal não é, necessariamente e literalmente, para todos. Revendo a definição cunhada por Mace: o design universal é o desenvolvimento de produtos e ambientes agradáveis esteticamente e usáveis por todas as pessoas, o **máximo possível** (Mace, 1998). Cabe aos designers e aos desenvolvedores, portanto, definir qual é o grau de universalidade deseja-se alcançar. Certamente, quanto maior for este grau, mais recursos (financeiro, pessoal, etc.) serão necessários a fim de acomodar no sistema as mais diferentes habilidades e necessidades que os usuários poderão ter.

Vale uma última ressalva de que há outros fatores que podem, de fato, ter um impacto negativo no acesso de qualquer pessoa aos recursos tecnológicos levantados por Zajicek e Edwards (2004), mas que não serão tratados nesta tese de doutorado. Um destes fatores é o econômico, pois nem todas as pessoas no mundo possuem condições financeiras para adquirir tecnologias. Outros fatores são o político, uma vez que há regimes conservadores que restringem o acesso e o uso das tecnologias pela população, e o cultural, já que as interfaces modernas apresentam referências culturais que podem ser verdadeiras barreiras para pessoas de diferentes culturas (Zajicek e Edwards, 2004).

3.3.1. Tecnologias assistivas

Em muitos casos, as pessoas com deficiência irão precisar de outros dispositivos alternativos ao teclado, ao mouse e ao monitor para conseguir utilizar um computador. Entram em cena então as chamadas tecnologias assistivas, ou ajudas técnicas, ou ainda auto-ajudas.

Conforme abordado anteriormente, as tecnologias assistivas são o conjunto de recursos que, de alguma maneira, contribuem para proporcionar às pessoas com

deficiências um maior nível de independência, de qualidade de vida ou de inclusão social, potencializando suas capacidades (Hogetop e Santarosa, 2001).

Não necessariamente uma tecnologia assistiva é um recurso computacional. Podem ser simples como uma bengala, uma cadeira de rodas, uma lupa ou um par de óculos, ou elaborados como teclados em Braille, sintetizadores de voz, sistemas de reconhecimento de fala e sistemas computadorizados para comunicação e controle interno de ambientes, envolvendo tanto hardware quanto software.

No caso dos recursos computacionais, existe uma grande diversidade de tecnologias assistivas. Provavelmente, a mais conhecida é o leitor de telas, que é um software que, utilizando um sintetizador de voz, realiza a "leitura" de todo o conteúdo textual da tela para cegos. Exemplos de leitores de tela atuais são o JAWS[3] e o Virtual Vision[4], ambos comerciais, e o NVDA[5], o Leitor de Telas do CPqD[6] e o Orca[7], estes três últimos desenvolvidos e distribuídos na forma de software livre. Além disso, existe ainda o DOSVOX, leitor de telas muito popular no Brasil, apesar de já obsoleto, desenvolvido pelo professor José Antonio Borges da Universidade Federal do Rio de Janeiro (UFRJ) (Borges, 2002a).

Outro exemplo de tecnologias assistivas são aquelas destinadas a pessoas com limitações motoras severas (como tetraplégicos, por exemplo), que impedem o uso do teclado e do mouse convencionais. Uma das mais simples tecnologias assistivas existentes para este público são os teclados virtuais com varredura. Basicamente, trata-se de um teclado apresentado na tela do computador no qual há um sistema de varredura pelas teclas e o usuário deve pressionar algum tipo de acionador para selecionar a tecla desejada. Há diferentes tipos de acionadores para uso com teclados virtuais, de acordo com as necessidades e habilidades dos usuários. Na Figura 3.1, podem ser vistos três tipos de acionadores: botões circulares, em diferentes tamanhos, pedal e um acionar baseado no sopro e sucção.

---

[3] http://www.freedomscientific.com/product-portal.asp. Informação capturada em 23/out/2009.
[4] http://www.virtualvision.com.br/. Informação capturada em 23/out/2009.
[5] http://www.nvda-project.org/. Informação capturada em 23/out/2009.
[6] http://www.mc.gov.br/projetos/leitor-de-telas. Informação capturada em 23/out/2009.
[7] http://projects.gnome.org/orca/. Informação capturada em 23/out/2009.

Figura 3. 1. Exemplos de acionadores para uso com teclados virtuais. À esquerda, botões circulares de diferentes tamanhos. Ao centro, um pedal. À direita, acionador baseado em sopro e sucção.

Normalmente, o layout apresentado no teclado virtual tende a ser o mesmo dos teclados convencionais (QWERTY) e a varredura é linear, ou seja, entre as linhas e colunas de teclas apresentadas no teclado virtual, fazendo com que o processo de escrita seja muito demorado – este é o caso do teclado virtual do Windows XP, que pode ser visto na Figura 3.2.

Figura 3. 2. Teclado Virtual do Windows XP. No caso, a varredura linear está passando exatamente pela letra "j" (em destaque).

Pensando nisso, um grupo de pós-graduandos em Ciência da Computação da UNICAMP, do qual também fiz parte, chegou a desenvolver o TFlex, um teclado que possibilita ao usuário alterar entre o layout alfabético e o QWERTY e apresenta dois modos de varredura: a linear e a binária, similar ao processo de busca binária em um vetor de dados (Juliato et al., 2004a; Juliato et al., 2004b).

De forma paralela aos teclados virtuais, há outras tecnologias para substituir o teclado tradicional. Uma delas são os programas de reconhecimento de voz, como o Via

Voice da IBM[8] e o projeto Motrix, também de autoria do professor Borges da UFRJ (Borges, 2002b).

Uma tecnologia que pode ser considerada ainda mais inovadora é a usada pelo Dasher (MacKay e Vertanen, 2005). O Dasher é uma interface para entrada de textos que utiliza um modelo de predição de palavras (em diversos idiomas, inclusive Português) para tornar a escrita eficiente. O software pode ser utilizado via mouse ou qualquer outro dispositivo de apontamento que substitua o mouse, como um rastreador ocular, que simplesmente move o cursor do mouse para o ponto na tela onde o usuário está olhando. A Figura 3.3 mostra uma tela do Dasher.

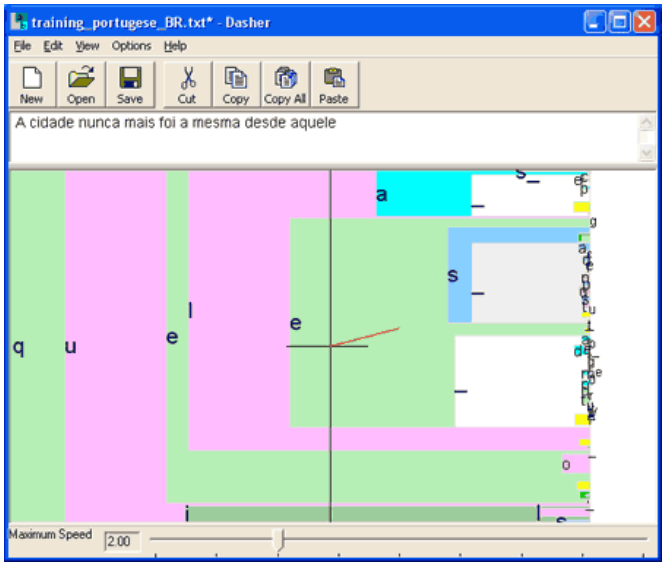


Figura 3. 3. Escrita com o Dasher.

Segundo os autores do Dasher, o uso do mesmo com um rastreador ocular possibilita escrever tão rápido quanto por digitação normal, sendo também mais rápido do que teclados virtuais.

Em termos de hardware, também existem ao menos dois tipos de teclados especiais: o teclado expandido e o teclado reduzido. O teclado expandido é um teclado cujas teclas são bem maiores do que as de um teclado convencional, facilitando o pressionamento das teclas por pessoas com transtornos que comprometem a precisão dos movimentos das

[8] http://www-01.ibm.com/software/pervasive/embedded_viavoice. Informação capturada em 23/out/2009.

mãos. Já o teclado reduzido é exatamente o oposto: trata-se de um teclado de tamanho diminuto, tanto é que suas teclas normalmente são pressionadas com o auxílio de uma pequena haste. O teclado reduzido foi desenvolvido para pessoas que apresentam uma amplitude de movimentos das mãos e dos braços muito reduzida, que não conseguiriam alcançar com facilidade todas as teclas do teclado convencional. A Figura 3.4 ilustra exemplos de teclados expandidos e reduzidos.

Figura 3. 4. Exemplo de teclados expandido (à esquerda) e reduzido (à direita).

Para usuários com coordenação motora muito reduzida, há a possibilidade de que o teclado expandido ainda não seja suficiente, ocasionando o pressionamento indesejável de mais de uma tecla ao escrever. Para superar esta dificuldade, há a possibilidade de se aplicar uma colméia por cima do teclado expandido. A colméia é uma "capa furada", geralmente produzida em plástico rígido, colocada por cima do teclado original. Cada furo da colméia situa-se exatamente acima de uma das teclas do teclado e, com isto, o usuário precisa colocar o dedo dentro de um único furo para digitar uma única letra por vez.

Apesar deste tópico não compreender todos os tipos de tecnologias assistivas existentes, é possível observar que cada tipo de tecnologia assistiva é dedicado a um público específico. Por exemplo, um leitor de telas é um sistema especialmente desenvolvido para usuários cegos. Assim, um raciocínio possível de ser feito é: se uma tecnologia assistiva é desenvolvida pensando-se em um público específico, então quer dizer que ela é **exclusiva, não inclusiva**, e, portanto, não pode ser considerada universal. Tal raciocínio é parcialmente correto, pois, de fato, o design universal não busca a concepção e o desenvolvimento de produtos exclusivos para um determinado público. Porém, o que se espera é que as tecnologias assistivas sejam, na medida do possível, **transparentes**, isto é, os sistemas que irão recebê-las não necessitarão de um (re)design especial para que tais

tecnologias funcionem de forma adequada. Para estes sistemas, é como se o usuário estivesse usando o teclado, o mouse e o monitor convencionais. No entanto, é notório que tecnologias como o leitor de telas quase sempre exigem algum (re)design específico, algo que é extensamente coberto por guias e recomendações de acessibilidade, que serão vistos a seguir.

Finalmente, de acordo com Choi et al. (2006), o preço das tecnologias assistivas é considerado elevado, motivo pelo qual as mesmas acabam ficando distantes dos seus potenciais usuários. Ainda, esclarecem que as tecnologias assistivas não devem ser consideradas como uma alternativa plausível para não se fazer um design universal – pelo contrário, as tecnologias assistivas são limitadas e muitas destas limitações podem ser superadas pelo design universal (Choi et al., 2006).

3.5. Guias de acessibilidade

A fim de facilitar o desenvolvimento de interfaces acessíveis, existem diversos guias e padrões de acessibilidade, principalmente em se tratando de páginas Web. As próximas sessões discutirão alguns dos principais guias e recomendações de acessibilidade.

3.4.1. WCAG 1.0

O *World Wide Web Consortium* (W3C) é o principal responsável por estabelecer padrões para a Internet, tais como o protocolo HTTP, a linguagem de marcação HTML e o formato de folhas de estilo CSS. Dentre os grupos de pesquisa e desenvolvimento do W3C, encontra-se a *Web Accessibility Initiative* (WAI), responsável por estabelecer *guidelines* para o desenvolvimento de conteúdo e ferramentas Web acessíveis.

No caso do conteúdo para a Web, o grupo da WAI elaborou as *Web Content Accessibility Guidelines* 1.0 (WCAG 1.0) (Chisolm et al., 1999). Trata-se de um documento

composto por 14 *guidelines* numeradas, cada uma contendo uma lista de *checkpoints* com níveis de prioridades variando de 1 (mais prioritário, o design **deve** satisfazê-lo, caso contrário um ou mais grupos de usuários não conseguirão acessar as informações no documento) a 3 (menos prioritário, o design **poderia** satisfazê-lo, caso contrário um ou mais grupos de usuários terão dificuldades para acessar as informações no documento). Tentou-se fazer cada *checkpoint* específico o suficiente de modo que se alguém revisar uma página Web ou um site poderá verificar se o *checkpoint* foi satisfeito ou não.

Dentre as recomendações presentes nas WCAG 1.0 estão *guidelines* solicitando fornecer alternativas para conteúdos visuais e auditivos através de equivalentes textuais, possibilitando o acesso às informações por usuários cegos (via leitores de tela) e surdos alfabetizados, não usar tabelas para o layout do documento, evitando assim problemas com o acesso a partir de leitores de tela, e garantir um conteúdo simples e claro, promovendo uma comunicação efetiva mesmo para aqueles com comprometimentos cognitivos ou com dificuldades de leitura.

O número de *checkpoints* em cada *guideline* varia, bem como os níveis de prioridade. A *guideline* 2, por exemplo, recomenda não confiar apenas em cores e possui 2 *checkpoints*: garantir que toda informação destacada por cores esteja também disponível sem cores (prioridade 1) e garantir que a combinação entre a cor principal e a cor de fundo da página tenha contraste suficiente (prioridade 2 para imagens e 3 para texto). Já a *guideline* 4, que solicita clareza na identificação da utilização de linguagem natural, possui 3 *checkpoints*: identificar claramente mudanças na linguagem de um texto (prioridade 1), especificar a expansão de cada abreviação ou acrônimo (prioridade 3) e identificar a linguagem principal do documento (prioridade 3).

Ainda no documento das WCAG 1.0, existe um tópico a respeito de níveis de conformidade. Uma página Web que cumpre com todos os *checkpoints* de nível de prioridade 1 é dita ter nível de conformidade “A” com as WCAG 1.0. Se, além de todos os *checkpoints* de prioridade 1, todos os *checkpoints* de prioridade 2 também forem satisfeitos, então a página Web terá nível de conformidade “AA” (ou “duplo-A”). Por fim, se todos os *checkpoints* de prioridades 1, 2 e 3 forem satisfeitos, a página Web terá nível de conformidade “AAA” (ou “triplo-A”).

3.4.2. ATAG 1.0

Além das WCAG, a W3C/WAI produziu outros dois conjuntos de *guidelines*, voltados para ferramentas de autoria para a Web e para "agentes de usuário", que são os softwares para acesso à Web, especialmente navegadores e tecnologias assistivas.

No caso das ferramentas de autoria para a Web, as *Authoring Tool Accessibility Guidelines* 1.0 (ATAG 1.0) (Treviranus et al., 2000) objetivam tanto auxiliar o design de ferramentas de autoria acessíveis quanto permitir que estas ferramentas produzam conteúdo acessível – basicamente, de acordo com as WCAG 1.0.

As sete *guidelines* presentes nas ATAG 1.0 são organizadas da mesma forma que as da WCAG 1.0, ou seja, cada uma possui uma lista de *checkpoints* a serem satisfeitos e cada *checkpoint* possui um nível de prioridade entre 1 e 3. Diferente das WCAG 1.0, em que a maioria dos *checkpoints* é bem específico, cada *checkpoint* das ATAG 1.0 possui uma descrição geral o suficiente para que os designers responsáveis pela ferramenta de autoria adotem a estratégia que for mais apropriada e conveniente para satisfazê-lo.

As quatro primeiras *guidelines* das ATAG 1.0 dizem respeito à produção de conteúdo acessível através da ferramenta de autoria ao passo que as três últimas são específicas da interface da ferramenta de autoria. Por exemplo, a *guideline* número 2 é "gerar marcação padrão", com a utilização de marcações da linguagem HTML definidas pela W3C em outros documentos, não necessariamente ligados apenas com acessibilidade, garantindo que os documentos na Web estejam com uma estrutura padronizada. Já a *guideline* número 7 é "garantir que a ferramenta de autoria é acessível a autores com necessidades especiais" e, inclusive, há uma nota lembrando que a documentação, o sistema de ajuda e a instalação são partes do software e precisam ser apresentados em um formato acessível também.

Dentro desta mesma *guideline* 7, há seis *checkpoints*. O primeiro *checkpoint* recomenda o uso de todos os padrões e convenções do sistema operacional e de acessibilidade aplicáveis. Para isto, é necessário seguir outras recomendações, de acordo com o tipo de software que está sendo desenvolvido e o sistema operacional a que ele se

destina. Os demais *checkpoints* desta *guideline* são para permitir ao autor editar todos os elementos e objetos de forma acessível, permitir alterar o modo de edição sem afetar a marcação e garantir a edição e a navegação entre os modos de edição acessíveis.

3.4.3. UAAG 1.0

O terceiro e último documento de acessibilidade da WAI/W3C, as *User Agent Accessibility Guidelines* 1.0 (UAAG 1.0) (Jacobs et al., 2002), diz respeito exatamente como navegadores e tecnologias assistivas podem entregar o conteúdo da Web de forma acessível aos usuários.

Como as WCAG 1.0 e as ATAG 1.0, as UAAG 1.0 também possuem a estrutura de *guidelines*, *checkpoints* e níveis de prioridade de 1 a 3 para cada *checkpoint*. Ao todo, doze *guidelines* compõem as UAAG 1.0. Se comparadas às WCAG 1.0 e às ATAG 1.0, uma grande diferença nas UAAG 1.0 é que as listas de *checkpoints* para cada *guideline* são ordenadas por nível de prioridade, ou seja, os *checkpoints* de prioridade 1 são apresentados antes dos de prioridade 2, que são apresentados antes dos de prioridade 3, o que facilita a navegação e a consulta ao documento.

Da mesma forma como as ATAG 1.0, as *guidelines* e *checkpoints* presentes nas UAAG 1.0 são descritas de uma maneira geral, dando liberdade aos desenvolvedores para encontrar uma forma de atendê-los. Por exemplo, a *guideline* 5 tem como título "Garantir ao usuário o controle do comportamento da interface" e tem cinco *checkpoints*, todos de prioridade 2. Um dos *checkpoints*, o 5.1, recomenda que não haja mudança de foco do conteúdo automática, enquanto outro *checkpoint*, o 5.5, recomenda a confirmação da submissão de formulários.

3.4.4. WCAG 2.0

A fim de atualizar as *guidelines* presentes nas WCAG 1.0[9], formou-se um grupo de trabalho no início de 2001 para elaborar uma nova versão, as WCAG 2.0 (Caldwell et al., 2008), oficialmente publicadas em sua versão final em dezembro de 2008. No geral, as *guidelines* e *checkpoints* da versão 2.0 ficaram mais alto nível, ou seja, possuem um maior nível de abstração, se comparadas com as da versão 1.0 que, de tão específicas, chegavam praticamente a referenciar como deveria ser feita a marcação em HTML em alguns casos.

A estrutura do documento que descreve e apresenta as WCAG foi alterada na versão 2.0, introduzindo-se mais um nível de abstração acima das *guidelines*, chamado de princípios. Ao todo, há quatro princípios nas WCAG 2.0:

1. Identificável (*Perceivable*) – a informação e os componentes da interface devem ser apresentados de maneira que os usuários possam identificá-los
2. Operável (*Operable*) – os componentes da interface e de navegação devem ser operáveis
3. Compreensível (*Understandable*) – a informação e a operação da interface devem ser compreensíveis
4. Robusto (*Robust*) – o conteúdo deve ser robusto o suficiente tal que ele possa ser interpretado por uma ampla variedade de dispositivos adotados pelo usuário, incluindo tecnologias assistivas

Dentro destes quatro princípios estão distribuídas doze *guidelines*. Adicionalmente, dentro de cada *guideline* há um ou mais *checkpoints*, agora chamados de critérios de sucesso (Caldwell et al., 2008). Por exemplo, dentro do princípio número 2, Operável, a primeira *guideline* é a 2.1, cujo título é "Teclado Acessível: faça todas as funcionalidades acessíveis por um teclado". Dentro da *guideline* 2.1, existem três critérios de sucesso, sendo que o primeiro deles, 2.1.1, foi nomeado simplesmente "Teclado" e recomenda que toda funcionalidade do conteúdo Web seja operável pelo teclado (Caldwell et al., 2008).

[9] Os outros documentos de acessibilidade da W3C, as ATAG e as UAAG, também passam por um processo de revisão no momento de escrita desta tese, mas não se encontram em sua versão final.

Enquanto os *checkpoints* das WCAG 1.0 apresentam um nível de prioridade de 1 a 3 diretamente relacionados com níveis de conformidade (A, AA e AAA), os critérios de sucesso das WCAG 2.0 possuem apenas níveis de conformidade, definidos entre A (mais baixo), AA e AAA (mais alto). Ou seja, de certa forma, pode-se dizer que os níveis de prioridade dos *checkpoints* da versão anterior (de 1 a 3) foram substituídos diretamente pelos níveis de conformidade (A, AA e AAA). Assim, por exemplo, uma página terá nível de conformidade "A" nas WCAG 2.0 se estiver de acordo com todos os critérios de sucesso de nível "A". A mesma página terá nível de conformidade "AA" se, além de atender os critérios de sucesso de nível "A", também atender os de nível "AA".

3.4.5. Por que as WCAG 2.0 ainda não deslancharam?

Apesar do trabalho extenso realizado para atualizar as WCAG, a adoção desta nova versão ainda não deslanchou até o momento da escrita desta tese de doutorado. De fato, nenhuma ferramenta de avaliação semi-automática listada no próprio site da W3C é descrita como aderente às WCAG 2.0 (WAI, 2009a) – as únicas conhecidas até o momento são a versão beta da ferramenta TAW (CTIC, 2009) e a AChecker (ATRC, 2009).

Uma justificativa plausível poderia ser pelo ainda pequeno tempo de existência das WCAG 2.0 – apenas oito meses se passaram desde a publicação final das WCAG 2.0 e a escrita desta tese de doutorado. Todavia, existem outros motivos tão ou mais pertinentes para justificar a baixa adesão atual.

Antes mesmo da publicação final das WCAG 2.0, o processo de atualização das *guidelines* foi duramente criticado. Clark (2007) chegou a escrever uma carta aberta a Tim Berners-Lee, criador da Web e diretor da W3C, solicitando o cancelamento das WCAG 2.0 ainda em 2007, quando as novas *guidelines* estavam em processo de análise e revisão.

Um dos principais pontos questionados é exatamente a quantidade de *guidelines* e o conjunto de documentos que compõem as WCAG 2.0, que continuam extensos (Clark, 2006; Popov, 2006; Lawson, 2006). Segundo comparação de Popov (2006), o tamanho do conjunto de documentos das WCAG 2.0 é equivalente à obra "Guerra e Paz" de Tolstoy.

Adicionalmente, outra crítica ferrenha feita é que, mesmo para pessoas fluentes em Inglês, o texto das WCAG 2.0 é de difícil leitura (Lawson, 2006) e, inclusive, violaria o próprio critério de sucesso 3.1.5 das WCAG 2.0 que diz respeito exatamente ao nível de leitura exigido para qualquer conteúdo na Web. Em outras palavras, a documentação das WCAG 2.0 não poderia ser considerada acessível se fosse rigorosamente avaliada pelas próprias WCAG 2.0!

Outras críticas frequentes são com relação à dificuldade de automatizar pelo menos parte da avaliação de acessibilidade, como ocorria com as WCAG 1.0, muito por conta da tentativa de se manter as *guidelines* em um nível de abstração maior, sem referências diretas a tecnologias usadas atualmente (Clark, 2006; Lawson, 2006). Outro ponto discutível nas WCAG 2.0 é que o próprio documento principal que apresenta as *guidelines* destaca que, mesmo se um site estiver em conformidade com todas as *guidelines*, pode ser que ele não seja acessível para todos os indivíduos, particularmente para aqueles com deficiências relacionadas à cognição e ao aprendizado (Caldwell et al., 2008). Além disso, em seu tópico a respeito de conformidade, o próprio documento principal esclarece que, para alguns sites, é impossível satisfazer os critérios de sucesso de nível AAA e, sendo assim, eles não são necessários (Caldwell et al., 2008). Ou seja, para um documento de *guidelines* que propõe padrões, é no mínimo contraditório e contra as próprias definições sobre o que é uma *guideline* (Free Dictionary, 2009b) afirmar que algumas são impossíveis de serem satisfeitas.

As discussões e críticas sobre as WCAG 2.0 chegaram a tal nível que um grupo formado por especialistas, pesquisadores e desenvolvedores Web, recusando-se a adotar as WCAG 2.0, elaborou uma atualização não-oficial das WCAG 1.0. Conhecido como WCAG Samurai (2008), o referido grupo lançou uma errata para as WCAG 1.0 e considerações práticas para se colocar em uso as *guidelines* da versão 1.0. Para Kelly et al. (2007), a simples presença do WCAG Samurai é um indicativo da falta de fé dos desenvolvedores Web nas WCAG 2.0.

Resumidamente, pode-se afirmar que as WCAG 2.0 enfrentam resistência de parte dos desenvolvedores e especialistas em acessibilidade Web e que, ainda por algum tempo, as WCAG 1.0 continuarão a ser adotadas como padrão de acessibilidade para conteúdo Web. Adicionalmente, outro ponto ainda pouco discutido é o impacto da atualização das

WCAG na legislação dos países que as adotaram como padrão de acessibilidade para sites de caráter governamental de interesse público (Termens et al., 2009). Algumas destas legislações serão apresentadas mais adiante no subtópico 3.5.

3.4.6. Por que o modelo de acessibilidade da W3C é falho?

Ainda que no futuro as WCAG 2.0 sejam aceitas pela comunidade de desenvolvedores Web tanto quanto atualmente são aceitas as WCAG 1.0, fortes questionamentos a respeito do modelo de acessibilidade da W3C vieram à tona nos últimos anos.

O modelo de acessibilidade da W3C é sustentado por três alicerces, ou três conjuntos de *guidelines*. O primeiro alicerce é exatamente as *guidelines* para conteúdo Web, as WCAG. As ATAG representam o segundo alicerce ao tratar de como as ferramentas de autoria para Web deveriam auxiliar a produção de conteúdo Web acessível, ou seja, conteúdo condizente com as WCAG. Finalmente, o terceiro alicerce são as UAAG, que tratam de como os navegadores e tecnologias assistivas podem fazer uso das WCAG para entregar conteúdo acessível aos usuários. O diagrama da Figura 3.5 mostra a relação entre os alicerces e entre conteúdo Web, ferramentas de autoria e navegadores e tecnologias assistivas.

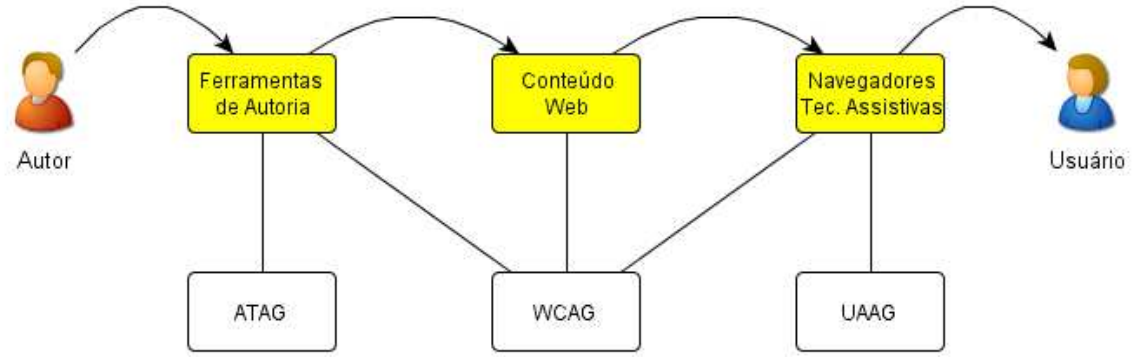


Figura 3. 5. Relação dos alicerces de acessibilidade da W3C.

Kelly et al. (2007) analisaram este modelo de acessibilidade da W3C. Segundo eles, para que um site na Web seja acessível de acordo com tal modelo, é preciso que o autor do conteúdo tenha gerado conteúdo acessível segundo as WCAG de alguma maneira, mesmo que sua ferramenta de autoria não seja completamente aderente às ATAG. Ainda, é preciso que o usuário final tenha à sua disposição um navegador e/ou tecnologias assistivas capazes de usufruir deste conteúdo em conformidade com as WCAG, isto é, o navegador e as tecnologias assistivas do usuário devem ter conformidade com as UAAG, o que nem sempre pode ser garantido uma vez que estas condições estão fora do controle dos desenvolvedores Web (Kelly et al., 2007).

É notável o papel central que as WCAG possuem no modelo de acessibilidade da W3C – sem dúvida alguma, as WCAG são o principal alicerce do modelo. E, assim sendo, as WCAG são de importância para todos, tanto para usuários quanto para desenvolvedores de tecnologias assistivas, desenvolvedores de navegadores e para autores de conteúdo Web. Adicionalmente, as WCAG deveriam refletir boas práticas para se obter um design acessível para todos na Web. No entanto, como salienta Kelly et al. (2007), não há força de evidência (isto é, indicativo de credibilidade e de validade) para as WCAG que garantam que a conformidade com elas resulte efetivamente em um design acessível para a Web. Além disso, completa Kelly et al., todo o processo de elaboração das WCAG, tanto da versão 1.0 quanto da versão 2.0, foi muito pouco transparente. Para Kelly et al., o processo adotado pela W3C deveria ser similar ao adotado no usability.gov (2006), no qual mais de 500 princípios de design foram extraídos de estudos previamente publicados a respeito de usabilidade e discutidos entre diversos especialistas, que avaliaram tanto a importância de cada um dos princípios quanto a força de evidência.

Outro aspecto levantado por Kelly et al. (2007) é quanto ao contexto do uso de uma aplicação na Web. Certamente, a acessibilidade em diferentes contextos (cultural, artístico, educacional, etc.) e para diferentes audiências pode e deve ter diferentes maneiras de ser trabalhada, o que não é compreendido pelas WCAG. Por exemplo, em um exercício no qual um professor propõe aos seus alunos a análise e a interpretação de uma gravura na Web, pode não ser apropriado ou pode ser uma tarefa muito difícil inserir um equivalente textual ou uma descrição para a gravura sem que isto influencie a própria interpretação dos alunos,

videntes ou não videntes. Ou a descrição textual da gravura para alunos não videntes pode não ser suficiente para que eles façam uma análise e interpretação da mesma forma que fariam os alunos videntes. Assim, para Kelly et al. (2007), o ideal seria ter outra forma na qual alunos com deficiências visuais consigam também participar deste tipo de atividade, mesmo que seja utilizando recursos que não estão na Web, como gravuras em alto relevo, por exemplo. Se as WCAG fossem levadas em consideração, no entanto, esta gravura na Web teria obrigatoriamente que possuir um equivalente textual associado.

Em outro artigo, Kelly et al. (2005) analisaram diversos estudos de acessibilidade de sites do Reino Unido e constataram que os níveis de conformidade com as WCAG 1.0 são bem decepcionantes, sendo que nenhum estudo relatou mais do que 6 % dos sites com nível de conformidade AA. Procurando por justificativas para este fato, extraíram de fóruns e listas de discussão as opiniões e queixas de desenvolvedores Web sobre as WCAG 1.0, condensadas na lista a seguir (Kelly et al., 2005):

- Natureza teórica das WCAG 1.0, que é unicamente baseada na visão da Web da W3C e ignora tecnologias de terceiros amplamente adotadas na Web, como Flash, GIF e PDF.
- Natureza fechada das WCAG 1.0. Por ter uma visão restrita às tecnologias e padrões definidos pela própria W3C, as soluções de acessibilidade propostas nas WCAG 1.0 são sempre baseadas no conteúdo da página Web e sequer considera possíveis funcionalidades de acessibilidade presentes no sistema operacional.
- Dependência de outras *guidelines* (ATAG e UAAG), conforme já explicitado anteriormente.
- Complexidade das *guidelines*. A compreensão das mesmas não é uma tarefa fácil e a quantidade de documentos para suportar as WCAG 1.0 dá uma idéia da dificuldade. Cooper e Rejmer (2001) também identificaram a dificuldade em encontrar informação nas WCAG 1.0 e navegar entre e nos documentos que circundam as WCAG 1.0.
- Falhas lógicas das *guidelines*. Não bastasse o texto complexo, existem falhas lógicas nas WCAG 1.0. Por exemplo, o *checkpoint* 11.1, de prioridade 2, recomenda

usar tecnologias W3C quando disponíveis e apropriadas para uma tarefa. Assim, ao invés de usar o formato GIF para imagens, que é proprietário e não definido pela W3C, uma página Web poderia ser obrigada a usar o formato PNG, que foi estabelecido pela W3C. Todavia, não existe qualquer evidência de que um determinado formato de imagem é melhor ou pior para a acessibilidade de uma página Web.

- Nível de entendimento sobre questões de acessibilidade. Nem todos os desenvolvedores ou Web designers são ou serão especialistas em acessibilidade e tecnologias assistivas, mas necessitam de guias claros e diretos sobre como fazer sites acessíveis. Trata-se de uma expectativa exagerada supor que qualquer um diretamente trabalhando no desenvolvimento de sites entenderá não apenas as WCAG, mas também como as WCAG interagem com outras *guidelines* da W3C e com tecnologias assistivas.

Alguns autores também apresentam que, para assegurar um bom nível de acessibilidade, apenas *guidelines* como as WCAG 1.0, que são baseadas em tecnologias, não são suficientes. É preciso também entender os usuários e como eles navegam pelos sites (Theofanos e Redish, 2003).

Finalmente, um fato curioso é que, atualmente, diversas empresas e organizações dão algum tipo de suporte à WAI e seus conjuntos de *guidelines*, como a Microsoft e a IBM. Entretanto, poucas delas adotaram as WCAG ou as ATAG em seus sites ou ferramentas de autoria para a Web.

3.4.7. Desafios ao lecionar sobre as WCAG

Além de todas as referências existentes a respeito das falhas nas WCAG, não se pode deixar de lado a minha própria experiência docente. Em quatro oportunidades distintas ao longo de meu doutorado, lecionei a respeito de design acessível e avaliação de acessibilidade na UNICAMP. No segundo semestre de 2007, no primeiro e no segundo

semestres de 2009, fui convidado pela titular da disciplina de graduação "Interfaces Humano-Computador", a professora doutora e minha orientadora Heloísa Vieira da Rocha, para lecionar um conjunto de aulas para alunos de graduação em Engenharia e Ciência da Computação. E, em 2008, ofereci uma Oficina de Acessibilidade para desenvolvedores envolvidos no projeto de Conteúdos Digitais do Ministério da Educação (Condigitais, 2009) – a maioria, de fato, também alunos de graduação dos cursos de Engenharia e Ciência da Computação da UNICAMP.

Nas quatro oportunidades, foi possível averiguar a dificuldade que desenvolvedores inexperientes em acessibilidade têm em tentar compreender as WCAG. Uma das dificuldades quase sempre encontrada é com relação à necessidade de se conhecer com alguma profundidade a linguagem de marcação HTML e folhas de estilo CSS, o que nem todos conheciam, apesar de serem alunos de dois dos melhores cursos de computação do país e, em alguns casos, até mesmo já estarem trabalhando em projetos de software para a Web. Outra dificuldade é que a maioria considerou que levaria muito tempo para verificar se uma página atende ou não a todos os *checkpoints* das WCAG 1.0, mesmo com o auxílio de ferramentas de verificação semi-automáticas (analisadas no subtópico 3.6.3 desta tese) – um dos alunos chegou a estimar, em sala de aula, que demoraria um dia para fazer a verificação completa das WCAG 1.0 em uma página Web.

Encarando todos os *checkpoints* das WCAG 1.0, um questionamento interessante, surgido durante as aulas do primeiro semestre de 2009, foi se existia alguma página efetivamente com nível de conformidade AAA, dada a complexidade e a quantidade de *checkpoints* a serem verificados. As próprias páginas do site da W3C não chegam a ter nível de conformidade AAA, nem mesmo a própria página em que estão publicadas as WCAG 1.0. Neste aspecto, segundo alguns pesquisadores, não é possível trabalhar com a maioria dos *checkpoints* de prioridade 3, tanto é que o WCAG Samurai simplesmente optou por excluí-los de sua errata não-oficial das WCAG 1.0 (WCAG Samurai, 2008).

As discussões com os alunos nas oportunidades em que lecionei sobre acessibilidade envolveram ainda quem é responsável por averiguar se uma página na Web efetivamente tem o nível de conformidade que ela própria afirma ter. Por exemplo, na página inicial do site do Ministério da Educação (MEC), há um selo atestando o nível de conformidade AAA, conforme pode ser visto na Figura 3.6. Contudo, uma análise rápida

feita com qualquer ferramenta de avaliação semi-automática terá como resultado a violação, pelo menos, dos *checkpoints* 11.2 (evitar funcionalidades desatualizadas da W3C), de prioridade 2, e 10.5 (incluir caracteres imprimíveis para separar links adjacentes, como uma barra vertical "|"), de prioridade 3. Assim, o nível de conformidade seria, no máximo, A.

Figura 3. 6. Página inicial do MEC[10]. No detalhe ampliado, no canto inferior esquerdo, o selo "AAA".

De acordo com a W3C (WAI, 2009b), os autores das páginas Web são os únicos responsáveis pelos usos de logotipos contendo níveis de conformidade (A, AA ou AAA) como aquele encontrado na página inicial do site do MEC. E, até onde se sabe, não há nenhuma organização ou indivíduo responsável formalmente por averiguar se os selos com

[10] A página inicial do MEC foi analisada e capturada em 18/ago/2009.

níveis de conformidade são usados de maneira apropriada pelos autores de páginas Web. Ou seja, não há a quem denunciar ou reclamar sobre o uso inapropriado de tais selos exceto aos próprios autores ou responsáveis pela página Web.

3.4.8. Outras *guidelines* e padrões de acessibilidade

Além dos conjuntos de *guidelines* de acessibilidade da W3C, outras entidades tem se empenhado nos últimos anos para estabelecer suas próprias *guidelines* e padrões. A maioria delas procura, pelo menos, alguma compatibilidade entre seus produtos e as tecnologias assistivas presentes no mercado.

3.4.8.1. *Guidelines* para Adobe Flash

O Adobe Flash é um padrão consolidado na Web para animações e multimídia. Porém, por ser um padrão comercial, não desenvolvido pela W3C, é sumariamente ignorado nas WCAG 1.0. Assim, havia uma dificuldade em se validar se um determinado conteúdo Web em Flash era ou não acessível.

Para tentar solucionar este problema, a Adobe vem trabalhando nos últimos anos em materiais que auxiliem Web designers a produzirem conteúdo acessível em Flash. E um dos resultados, publicado no site da Adobe, é um conjunto de doze *guidelines* que sumarizam as melhores práticas para o design com Flash (Adobe, 2009).

As *guidelines* da Adobe para Flash possuem uma descrição em alto nível e oito das doze *guidelines* incluem links para documentos com explicações e exemplos de como colocá-las em prática, numa tentativa de facilitar sua adoção por Web designers.

3.4.8.2. *Guidelines* para Java

A linguagem de programação Java conta com classes e bibliotecas específicas para oferecer acessibilidade. Baseado nestas classes e bibliotecas e em algumas práticas de programação para oferecer melhor acessibilidade, Schwerdtfeger (2006), pesquisador da IBM, propôs um conjunto de *guidelines* de acessibilidade para se desenvolver software 100% Java. O conjunto de *guidelines* de Schwerdtfeger é tão completo que é referenciado até mesmo pela própria Sun, desenvolvedora da linguagem Java.

As *guidelines* de Schwerdtfeger (2006) podem ser divididas em dois grandes grupos. No primeiro grupo, estão presentes *guidelines* relacionadas às "práticas essenciais de programação para acessibilidade", usando os componentes e classes já existentes em Java. Por exemplo, a primeira *guideline* tem como título "Faça sua aplicação acessível pelo teclado" e, assim como todas as outras *guidelines*, possui como exemplo trechos de código que atendem a *guideline* e como fazer para testar se o programa em Java atende ou não a *guideline*. No caso específico da primeira *guideline*, o teste sugerido é tentar usar o programa apenas com o teclado, sem o mouse.

O segundo grupo de *guidelines* de Schwerdtfeger (2006) diz respeito a como um desenvolvedor Java pode criar seus próprios componentes e classes acessíveis em Java. Resumidamente, e para tal, é preciso implementar a interface "*Accessible*" da linguagem Java.

3.4.8.3. *Guidelines* IBM

Além das *guidelines* de acessibilidade de Schwerdtfeger (2006) para software 100% Java, outros pesquisadores da IBM também conceberam outros conjuntos de *guidelines* de acessibilidade (IBM, 2008). Na forma de *checklists*, estas *guidelines* da IBM são agrupadas em: software, conteúdo Web, Java, aplicações para o Lotus Notes, hardware (periféricos, como teclado, mouse, entre outros), hardware autocontido (impressoras, copiadoras, aparelhos de fax e similares) e documentação. Pode-se ver que a IBM procurou abordar

todos os elementos que compõem um sistema, não apenas o conteúdo Web, incluindo o hardware e a documentação, diferente das demais organizações que propuseram *guidelines* de acessibilidade.

Um exemplo de item presente no *checklist* de hardware, e especialmente importante para qualquer tipo de teclado usado por pessoas com deficiências visuais, é fornecer teclas que possam ser identificadas através do tato sem que haja a necessidade de ativá-las. Em geral, os teclados de computadores pessoais que adotam o layout QWERTY já trazem ao menos uma indicação em alto relevo de onde se encontram as teclas das letras "F" e "J", nas quais devem permanecer os dedos indicadores das mãos.

Outros exemplos de *guidelines*, desta vez para a documentação, demandam que toda a documentação do software também deve estar em um formato acessível, e que as funcionalidades relacionadas com acessibilidade devem estar documentadas.

Por fim, as *guidelines* da IBM para a Web são muito similares às presentes nas WCAG 1.0. Por exemplo, para imagens e animações, há uma *guideline* recomendando o uso de equivalentes textuais por meio do atributo "alt" da linguagem de marcação HTML.

3.4.8.4. *Guidelines* GNOME para GNU/Linux

Não apenas empresas e produtos comerciais estão oferecendo suporte a designers e desenvolvedores para acessibilidade. Para GNU/Linux, também existem recursos de acessibilidade mantidos pela comunidade de software livre.

No ambiente gráfico GNOME para GNU/Linux há uma biblioteca chamada *Accessibility Toolkit* (ATK), cuja função é permitir a comunicação entre aplicativos no ambiente GNOME e tecnologias assistivas. Ou seja, a biblioteca ATK oferece um nível básico de acessibilidade para aplicações do GNOME. Para garantir um nível mais alto de acessibilidade, é preciso que desenvolvedores e designers de software para GNOME tomem algumas ações.

Assim, um conjunto de *guidelines* foi publicado e separado em duas partes (Alexander et al., 2008). A primeira parte trata de recomendações de codificação para

oferecer acessibilidade nos programas que executam no ambiente GNOME. Por exemplo, uma das *guidelines* recomenda o uso de uma função da biblioteca ATK para adicionar descrição textual para imagens e ícones da aplicação. A segunda parte apresenta recomendações de mais alto nível e que podem ser aplicáveis a qualquer aplicação GNOME. Por exemplo, há uma *guideline* que recomenda não usar cores como a única maneira de distinguir informação – como no caso de campos obrigatórios de um formulário.

3.6. Leis de acessibilidade

Em diversos países já existem leis para orientar o design acessível de sites, principalmente os governamentais e os de interesse público.

Desde 1998, todos os sites e tecnologias de informação das agências federais dos Estados Unidos são obrigados a estar em conformidade com a *Section 508*, uma lei que impõe padrões técnicos e critérios funcionais visando acessibilidade (Section 508, 2000).

Na Europa, os países membros da União Européia foram encorajados a adotar as *guidelines* da W3C durante o processo de desenvolvimento de ferramentas de autoria e de conteúdo para a Web, principalmente em se tratando de sites do setor público (Belder, 2002). Adicionalmente, tal resolução também solicita aos governos que se esforcem para disponibilizar tecnologia, treinamento e suporte às pessoas com deficiências. Porém, reconhece que há diferenças nos países membros da União Européia com relação à forma de promoção de acessibilidade na Internet.

A Austrália não possui uma lei que obriga sites governamentais a seguirem determinadas *guidelines*. No entanto, o *Australian Human Rights and Equal Opportunity Commission* formalmente endossa as WCAG 1.0 como padrão de acessibilidade para a Web (HREOC, 2009).

Finalmente, no Brasil, por meio do Ministério do Planejamento, o governo desenvolveu um conjunto de *guidelines* baseado nas WCAG 1.0 para guiar o design de seus próprios sites (Governo Eletrônico, 2005). Nomeado "Modelo de Acessibilidade de

Governo Eletrônico", o documento tem como um diferencial recomendar a aplicação de testes com usuários com deficiências sempre que possível, enquanto outras regulamentações pelo mundo não estabelecem ou sequer sugerem como deve ser feita tal validação da acessibilidade.

3.7. Avaliação de Acessibilidade

Assim como para usabilidade, há métodos de inspeção e empíricos para avaliar a acessibilidade. Porém, a maioria dos métodos de avaliação de acessibilidade atuais destina-se exclusivamente a avaliar o conteúdo de páginas web, uma vez que são baseados na conformidade com certas *guidelines* de acessibilidade, principalmente com as WCAG 1.0 e as recomendações da *Section 508*. Alguns dos principais métodos de avaliação de acessibilidade serão apresentados a seguir.

3.6.1. Conformidade com *guidelines*

A avaliação de conformidade com *guidelines* é um método muito simples de inspeção da interface em busca de problemas de acessibilidade. Trata-se de um método muito similar à avaliação heurística, exceto pelo fato de que, quase sempre, o número de *guidelines* é bem elevado (e também pelas diferenças existentes entre os conceitos de heurística e de *guideline* vistos no capítulo 1): dado um conjunto de *guidelines*, como as WCAG 1.0, o avaliador, que preferencialmente deve ser um especialista na área, verifica se a página ou site em avaliação está de acordo com cada *guideline* e *checkpoint* do conjunto (Lazar, 2001). Caso não esteja, descreve as violações existentes em um relatório, incluindo uma justificativa para cada violação.

Como todo método de inspeção de interfaces, a verificação de conformidade com *guidelines* não requer a presença de usuários durante o processo de avaliação. Porém, assim

como ocorre com a usabilidade, este tipo de avaliação não substitui os testes com usuários – de fato, pode-se dizer que os métodos são complementares.

3.6.2. Por que a avaliação de acessibilidade com as WCAG nem sempre é apropriada?

O método mais popular atualmente para se avaliar a acessibilidade de um site é através da verificação de conformidade com as WCAG 1.0. A existência de uma grande quantidade de ferramentas para auxiliar na verificação de conformidade com as WCAG 1.0 – há mais de 100 oficialmente listadas pela W3C (WAI, 2009a) – contribui e muito para esta popularização.

Apesar de muitas vezes serem chamadas de "ferramentas de avaliação automáticas de acessibilidade", nem todos os *checkpoints* das WCAG 1.0 são possíveis de serem verificados automaticamente. Cooper e Rejmer (2001) fizeram uma análise de uma destas ferramentas e verificaram que apenas cerca de 20% dos *checkpoints* das WCAG 1.0 podem ser automaticamente verificados. Ou seja, mesmo com o auxílio de uma destas ferramentas, um avaliador humano ainda é necessário para verificar a conformidade com a grande maioria dos *checkpoints* das WCAG 1.0. Assim, a maneira mais apropriada de denominar estas ferramentas seria de "semi-automáticas" e não "automáticas".

Como a verificação de conformidade humana é sempre necessária, todas as dificuldades ao se trabalhar com as WCAG 1.0 anteriormente analisadas são aplicáveis neste tipo de avaliação de acessibilidade Web: natureza teórica e fechada das WCAG 1.0, dependência de outras *guidelines*, complexidade e falhas lógicas das *guidelines* (Kelly et al., 2005). Adicionalmente, Leporini e Paternò (2004) consideram os relatórios gerados pelas ferramentas semi-automáticas difíceis de serem entendidos por desenvolvedores inexperientes.

Não bastassem estes fatos, um avaliador humano que adota uma destas ferramentas semi-automáticas deve sempre ter atenção redobrada para os relatórios de acessibilidade produzidos, visto que algumas geram em seus resultados falsos positivos e/ou falsos

negativos. Um falso positivo ocorre quando a ferramenta semi-automática relata a violação de um *checkpoint* quando, na verdade, ele não é violado. Já um falso negativo acontece quando a ferramenta não relata corretamente a violação de um determinado *checkpoint* quando ele efetivamente é violado. Assim, se o avaliador não confia cegamente na ferramenta semi-automática de verificação de conformidade com as WCAG 1.0 que utiliza eventualmente ele terá que verificar todos os *checkpoints* manualmente!

Por fim, segundo Brajnik (2006), *guidelines* como as WCAG 1.0 não ajudam o avaliador a distinguir problemas importantes de triviais, mesmo com níveis de prioridade definidos. Por exemplo, um dos *checkpoints* das WCAG 1.0 solicita que toda imagem na página Web possua um texto alternativo, sendo um *checkpoint* de prioridade 1. Todavia, de acordo com Brajnik, a maioria das imagens na Web tem propósito "emocional" e não agregam conteúdo ao documento – portanto, não há a necessidade de efetivamente incluir um texto alternativo em todas as imagens.

3.6.3. Estudo sobre as ferramentas semi-automáticas

Para analisar a real eficácia das ferramentas semi-automáticas e das WCAG 1.0 na avaliação de acessibilidade Web, sentimos a necessidade de realizar testes com usuários/potenciais avaliadores de acessibilidade. Assim, quatro das mais representativas ferramentas semi-automáticas em termos das funcionalidades oferecidas foram selecionadas para serem testadas com usuários:

- Truwex (Erigami, 2009): é uma ferramenta online de verificação de conformidade com as WCAG 1.0 e com as Section 508, capaz de gerar relatórios na forma de textos e de gráficos, chamados na ferramenta de "mapa", indicando os problemas diretamente na página Web. As Figuras 3.7 e 3.8 ilustram algumas telas da Truwex.

Truwex Online 2.0: Section 508 and WCAG Accessibility, Privacy, Quality Assurance Tool

Erigami Home | Accessibility Check | BITV Check | Google Analytics Check | Help | Contacts | Test Results

What is Truwex? Read Truwex Factsheet.
Wish Truwex check your entire website? Download Truwex 2.0 trial version.
Buy Truwex SMB for only $390

Check another page

Page URL: http://www.comvest.unicamp.br

Properties | Issues | Map | Inventory | Profile

Accessibility

7 issues in 18 instances

[508 (A)] [WCAG 1.1 (1)] Alt is missing 3

Description. Alt is missing (image, applet)

Elements with missing alt

Element type	Source URL
IMAGE	http://www.comvest.unicamp.br/images/barra_inicial.gif
IMAGE	http://www.comvest.unicamp.br/images/barra_submenu.gif
IMAGE	http://www.comvest.unicamp.br/images/seta.gif

View the Issue details on a web page Map.

[508 (N)] [WCAG 12.4 (2)] Form control without explicit label is found 1

[WCAG 2.2 (2)] Low-contrast text is found 1

[WCAG 3.2 (2)] Doctype missing 1

[WCAG 3.4 (2)] Fixed fonts are found 9

[WCAG 4.3 (3)] Document language markup is missing 1

[WCAG 13.1 (2)] Link name/destination clash 2

13 warnings in 40 instances

[508 (C)] [WCAG 2.1 (1)] The use of color must not be a single method for indicating important information on a web page 1

[508 (D)] [WCAG 6.1 (1)] Documents must be organized so they are readable without requiring user agent 1

Figura 3. 7. Relatório textual da ferramenta Truwex.

Truwex Online 2.0: Section 508 and WCAG Accessibility, Privacy, Quality Assurance Tool

Erigami Home | Accessibility Check | BITV Check | Google Analytics Check | Help | Contacts | Test Results

Page URL: http://www.comvest.unicamp.br

Properties | Issues | Map | Inventory | Profile

Click on issue number to highlight it on map

Map | DHTML source

Show arrow

Accessibility issues

[508 (A)] [WCAG 1.1 (1)] Alt is missing
1 2 3 All

[508 (N)] [WCAG 12.4 (2)] Form control without explicit label is found
1

[WCAG 2.2 (2)] Low-contrast text is found
1

[WCAG 3.4 (2)] Fixed fonts are found
1 2 3 4 5 6 7 8 9 All

[WCAG 13.1 (2)] Link name/destination clash
1 2 All

Accessibility warnings

[508 (D)] [WCAG 6.1 (1)] Documents must be organized so they are readable without requiring user agent support for style sheets
1 2 All

[WCAG 2.2 (2)] Images must have sufficient contrast
1 2 3 4 5 6 7 8 9 All

[WCAG 3.1 (2)] Use markup instead of images where

COMVEST
Comissão Permanente para os Vestibulares

Busca

UNICAMP

Vestibular 2009 | Estatísticas | Anos Anteriores | PAAIS | Vagas Re

Inscrições | Manual | Isenção

[508 (N)] [WCAG 12.4 (2)] Form control without explicit label is found
Input id:
Input type: Text box

Comvest abre inscrições para a Oficina sobre a Redação no Vestibular Unicamp

A Comissão Permanente para os Vestibulares da Unicamp está disponibilizando o formulário de inscrição para a oficina "A Redação no Vestibular Unicamp". As inscrições poderão ser feitas até as 17 horas do dia 17 de abril, exclusivamente pela internet, nesta página. A oficina será realizada em quatro sábados do mês de maio (09, 16, 23 e 30/05), das 08 às 13 horas, no campus da Unicamp, em Campinas. Este ano serão oferecidas 360 vagas. Leia mais

Fique atento!

Lista de Livros 2010
Confira aqui a nova relação de livros para o Vestibular 2010

Vestibular 2009
Preencha aqui o questionário dos matriculados

Vestibular 2009
Comvest encerra as chamadas

Vestibular 2009
Unicamp divulga notas e classificação dos candidatos

Notícias e Clipping

Confira como foram as Provas de Aptidão do Vestibular 2009

Comvest lança livro com as melhores Redações do Vestibular Unicamp 2008 e homenageia jovens autores

Figura 3. 8. Relatório gráfico da ferramenta Truwex, indicando o local na página onde há uma violação de *checkpoint* das WCAG 1.0.

- DaSilva (Acessibilidade Brasil, 2009): assim como a Truwex, a DaSilva é uma ferramenta online para verificação de conformidade com as WCAG 1.0. Desenvolvida no Brasil, a DaSilva é a única ferramenta no momento da escrita desta tese capaz de gerar relatórios de acessibilidade de acordo com as *guidelines* propostas pelo governo brasileiro. Na Figura 3.9, é possível verificar uma tela de um relatório gerado pela DaSilva.

Figura 3. 9. Relatório de violações de *checkpoints* da ferramenta DaSilva.

- HTML *Validator for Firefox* (Gueury, 2009): como o nome sugere, trata-se de uma ferramenta que funciona como um complemento para o navegador Firefox. A ferramenta, na verdade, incrementa a visão do código-fonte de uma página Web no Firefox, indicando (no código-fonte) potenciais problemas de acessibilidade de acordo com as WCAG 1.0. O diferencial da ferramenta fica por conta da uma funcionalidade de "limpeza" do código, que possibilita corrigir automaticamente

alguns dos problemas de acessibilidade encontrados. Um exemplo de relatório de uso da ferramenta pode ser visto na Figura 3.10.

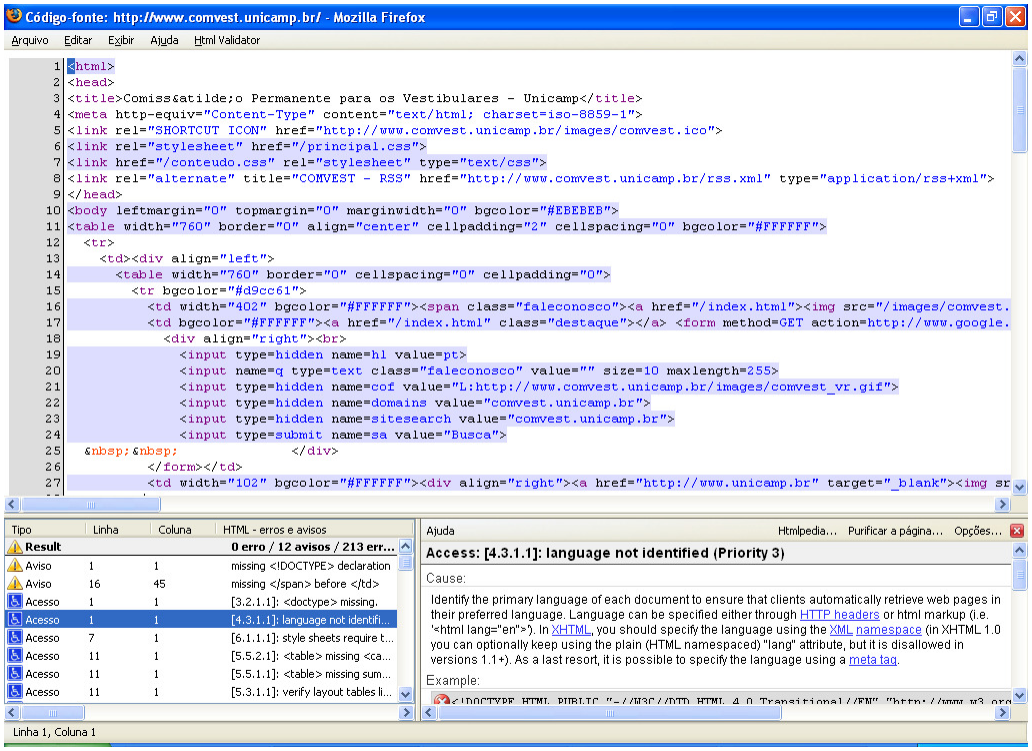
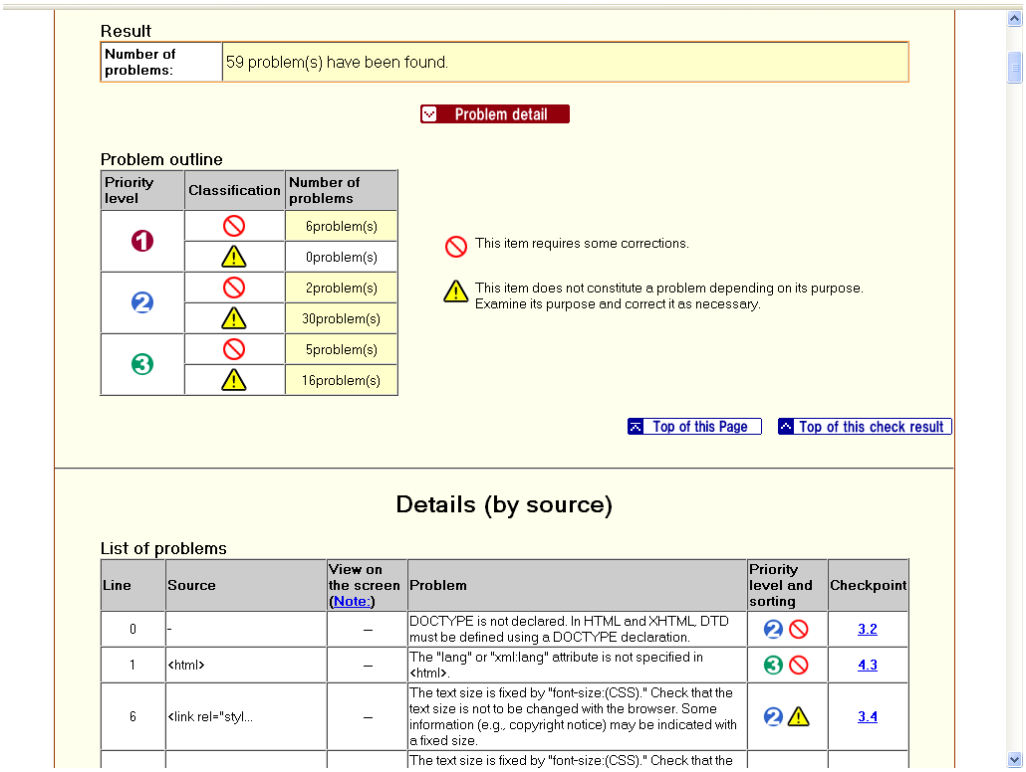


Figura 3. 10. Relatório de acessibilidade da ferramenta HTML Validator for Firefox.

- *Fujitsu Web Accessibility Inspector* (Fujitsu, 2009): da mesma forma como a Truwex, a ferramenta de inspeção de acessibilidade da Fujitsu é capaz de gerar relatórios textuais e gráficos, seguindo como referência as WCAG 1.0. Ainda, como é uma aplicação do tipo "*stand-alone*" para Windows, pode ser usada para verificar tanto conteúdo na máquina local quanto na Web. O formato dos relatórios da ferramenta é configurável pelo avaliador, que pode escolher quais checkpoints devem ser considerados na avaliação, como ordená-los e agrupá-los (por nível de prioridade ou pela sua ordem numérica). Na Figura 3.11, pode-se ver um relatório gerado pela ferramenta.



Result

Number of problems:	59 problem(s) have been found.

Problem detail

Problem outline

Priority level	Classification	Number of problems
1		6problem(s)
		0problem(s)
2		2problem(s)
		30problem(s)
3		5problem(s)
		16problem(s)

This item requires some corrections.

This item does not constitute a problem depending on its purpose. Examine its purpose and correct it as necessary.

Top of this Page | Top of this check result

Details (by source)

List of problems

Line	Source	View on the screen (Note:)	Problem	Priority level and sorting	Checkpoint
0	-	–	DOCTYPE is not declared. In HTML and XHTML, DTD must be defined using a DOCTYPE declaration.	2	3.2
1	<html>	–	The "lang" or "xml:lang" attribute is not specified in <html>.	3	4.3
6	<link rel="styl...	–	The text size is fixed by "font-size:(CSS)." Check that the text size is not to be changed with the browser. Some information (e.g., copyright notice) may be indicated with a fixed size.	2	3.4
			The text size is fixed by "font-size:(CSS)." Check that the		

Figura 3. 11. Relatório gerado pela ferramenta Fujitsu Web Accessibility Inspector.

Foram recrutados três mestres em ciência da computação para participar dos testes das ferramentas semi-automáticas de acessibilidade. Antes de iniciar os testes, entretanto, todos os participantes receberam um termo de consentimento, descrevendo os objetivos do estudo e assegurando que dados pessoais seriam mantidos em sigilo, e um questionário para levantar a experiência e familiaridade de cada um com usabilidade e acessibilidade. Ambos os documentos podem ser visualizados no Anexo A.

As respostas dos participantes ao questionário indicam que todos eram experientes avaliadores de usabilidade com pouca ou nenhuma experiência em avaliação de acessibilidade. Porém, todos possuíam uma boa idéia do que é acessibilidade. Adicionalmente, todos os participantes trabalhavam ou já haviam trabalhado em projetos nos quais a acessibilidade tinha sido abordada de alguma forma, mesmo que não formalmente presente como um requisito de qualidade de seus projetos.

Cada ferramenta foi testada individualmente por cada um dos participantes, que tinham como tarefa verificar a conformidade de todos os *checkpoints* das WCAG 1.0 para

uma página pré-selecionada em um período de, no máximo, uma hora, podendo desistir ou parar o teste a qualquer momento que desejassem e por qualquer razão. Em cada sessão de teste, o experimentador (no caso, eu) foi responsável por anotar os comentários, reações e dificuldades dos participantes ao usarem as ferramentas e acessarem as WCAG 1.0. Ao final de cada sessão de teste, os participantes foram convidados a preencher outro questionário para se obter suas impressões a respeito da ferramenta testada, como pode ser visto no Anexo A.

Como os participantes tinham pouca experiência com acessibilidade, no início da primeira sessão de testes de cada um deles foi realizada uma apresentação informal sobre acessibilidade e as WCAG 1.0: como as *guidelines* estão organizadas, como navegar por entre os documentos e como verificar o nível de prioridade de cada *checkpoint*.

Para tentar reduzir a influência da ordem das ferramentas na opinião dos participantes, cada um começou e terminou os testes com uma ferramenta diferente, como visto na tabela 3.2. Também, uma página diferente foi usada para cada ferramenta, como descrito na tabela 3.3. É importante destacar que as páginas selecionadas para os testes são uma amostra representativa do que se pode encontrar no universo de páginas Web: textos, formulários, recursos de animação, vídeo e áudio e scripts Java.

Tabela 3. 2. Ordem de testes das ferramentas, por participante.

Participante	Ordem das Ferramentas
P1	Truwex – DaSilva – HTML Validator – Fujitsu
P2	DaSilva – HTML Validator – Fujitsu – Truwex
P3	Fujitsu – Truwex – DaSilva – HTML Validator

Tabela 3. 3. Páginas analisadas com cada ferramenta.

Ferramenta	Página analisada
DaSilva	Comvest (Podcasts)[11]
Fujitsu	TV Senado[12]
HTML Validator	Nações Unidas[13]
Truwex	Receita Federal (Consulta de CPF)[14]

Na última sessão de testes de cada participante, um último questionário foi apresentado para que pudessem expressar opiniões finais a respeito das ferramentas avaliadas, das WCAG 1.0 e sobre o estudo em si. O questionário final pode ser visto no Anexo A.

3.6.3.1. Resultados do estudo sobre as ferramentas semi-automáticas

Em geral, os participantes aprovaram o "mapa" mostrado na ferramenta Truwex, que permite ao avaliador visualizar na própria página Web as violações de *checkpoints*. No entanto, alguns *checkpoints* não são possíveis de serem analisados visualmente no "mapa" da ferramenta Truwex, apesar desta opção estar disponível para todos os *checkpoints*, o que resultou em muita frustração para um dos participantes. Também, outro participante ficou em dúvida sobre o que exatamente seria um *issue* e o que seria um *warning* (rótulos que aparecem no resumo da verificação, indicando um problema encontrado automaticamente e um *checkpoint* a ser analisado manualmente, respectivamente) e qual seria a relação de ambos com a gravidade dos problemas de acessibilidade encontrados.

[11] http://www.comvest.unicamp.br/podcast/podcast.html. Informação capturada em 03/nov/2009.
[12] http://www.senado.gov.br/tv/. Informação capturada em 03/nov/2009.
[13] http://www.un.org/. Informação capturada em 03/nov/2009.
[14] http://www.receita.fazenda.gov.br/Aplicacoes/ATCTA/cpf/ConsultaPublica.asp. Informação capturada em 03/nov/2009.

Ainda, dois participantes sentiram falta de visualizar também o código fonte da página Web em avaliação e um deles afirmou que a ferramenta Truwex é uma opção mais indicada para aqueles que não conhecem HTML. Outra queixa feita por todos os participantes nos testes com a ferramenta Truwex foi com relação à dificuldade em se identificar o nível de prioridade de cada um dos *checkpoints* sendo analisados, ou seja, identificar a severidade de cada um dos problemas encontrados. A informação do nível de prioridade dos *checkpoints* é sempre mostrada logo após o número do *checkpoint* e entre várias outras informações, por exemplo: [508 (A)] [WCAG 1.1 (1)], indica que o *checkpoint* 1.1 das WCAG 1.0, equivalente à *guideline* A da *Section 508*, é de prioridade 1, valor que aparece entre parênteses logo após o número do *checkpoint*.

Com relação às WCAG 1.0, durante os testes com a Truwex um dos participantes achou muito confuso ter que navegar pela documentação das *guidelines* e reclamou sobre a falta de clareza do *checkpoint* 13.6, cuja recomendação é para que links relacionados sejam agrupados e que tais grupos sejam identificados. Ainda, outro participante não entendeu a relação entre a marcação específica de rótulos (*labels*) para campos de formulários e acessibilidade, conforme recomendação dos *checkpoints* 10.2 e 12.4.

Já na ferramenta DaSilva, os participantes consideraram relevante a funcionalidade de verificar os *checkpoints* diretamente nas linhas do código, embora dois (P1 e P3, que já haviam participado dos testes da Truwex) terem sentido falta de uma funcionalidade como o "mapa" da Truwex. E, por não terem familiaridade com o código HTML da página Web avaliada, todos os participantes tiveram alguma dificuldade inicial em navegar pelo código. Ainda assim, os participantes aprovaram o agrupamento dos *checkpoints* por nível de prioridade na ferramenta daSilva, facilitando identificar a gravidade de cada *checkpoint*. Uma dificuldade que também foi constatada foi com relação à ferramenta daSilva não informar em seu relatório qual conjunto de *guidelines* foi usado: se as WCAG 1.0 ou se as do governo brasileiro (E-GOV), tanto é que um dos participantes, por desatenção, não havia selecionado as WCAG 1.0 e sentiu-se perdido, uma vez que a numeração dos *checkpoints* das E-GOV é diferente das WCAG 1.0, embora tenham praticamente o mesmo conjunto de *guidelines* e *checkpoints*. Finalmente, dois participantes sugeriram que o relatório gerado pela daSilva poderia ser alterado de acordo com as preferências do avaliador, possibilitando

ordenar de forma diferente (por quantidade de ocorrências, por número do *checkpoint*, entre outras).

Provavelmente, como a ferramenta daSilva apresenta os *checkpoints* de maneira textual apenas, sem uma visualização gráfica, os participantes se fixaram mais nas WCAG 1.0 do que nas funcionalidades oferecidas pela ferramenta, resultando em mais queixas dos *checkpoints* das WCAG 1.0, em especial dos *checkpoints* 3.2, a respeito do uso de gramáticas formais publicadas (HTML, CSS, entre outras), 13.4, a respeito do uso de mecanismos de navegação consistentes por todo o site, 5.2, sobre como deve ser a formatação de tabelas com dois ou mais níveis lógicos, e 8.1, que trata da necessidade de elementos programáveis como *scripts* e *applets* serem acessíveis por tecnologias assistivas. Inclusive, um dos participantes achou um "exagero" a proposta do *checkpoint* 11.4, que diz que, se apesar de todos os esforços, não for possível fazer a página acessível, o desenvolvedor deveria criar outra página com conteúdo equivalente e acessível.

A outra ferramenta apenas textual, a *HTML Validator* teve como principais qualidades destacadas pelos participantes a possibilidade de se corrigir alguns dos problemas automaticamente, a rapidez na geração do relatório de conformidade com as WCAG 1.0 e as justificativas e os exemplos de erros incluídos em cada *checkpoint* violado e/ou a ser verificado manualmente. No entanto, todos demonstraram dificuldades em navegar pela lista de problemas, que apresentava tanto problemas de acessibilidade quanto no código HTML – segundo um dos participantes, é "chato ter tudo junto". Outro participante considerou estranho que os resultados fossem mostrados na mesma janela de exibição do código fonte no navegador Firefox. Também, assim como ocorreu com a daSilva, dois participantes manifestaram desejo em organizar de outras formas o relatório de problemas – porém, a ferramenta *HTML Validator* só apresenta um tipo de listagem.

Com relação às dificuldades com as WCAG 1.0, um dos participantes não entendeu o que seria um *flicker*, descrito no *checkpoint* 7.1. Outro *checkpoint* que causou questionamentos em dois participantes foi o 5.3, que recomenda não usar tabelas para layout da página. Para um destes participantes, falta material nas WCAG 1.0 ou na própria ferramenta de verificação de conformidade para auxiliar na transformação de tabelas em outros elementos para formatar o layout da página. Outro *checkpoint*, o 6.3, que diz que a página deveria ser usável mesmo quando *scripts* e *applets* fossem desligados ou não

suportados, também foi considerado muito difícil de ser corrigido por um dos participantes, que também reclamou da ausência de material para auxiliar neste tipo de problema. O mesmo participante também não conseguiu entender porque o uso de uma marcação como "center", considerada depreciada pela W3C, mas normalmente interpretada corretamente por praticamente todos os navegadores Web, é um problema de acessibilidade segundo *checkpoint* 11.2 das WCAG 1.0.

Na última ferramenta avaliada, a Fujitsu, os participantes foram unânimes em elogiar a funcionalidade "view", que é muito parecida com o "mapa" da Truwex. Também, assim como a *HTML Validator*, o tempo para aguardar o relatório ficar pronto foi bem menor do que as outras ferramentas, uma vez que a Fujitsu é uma aplicação *stand-alone*, executada diretamente da máquina local, sem a necessidade de encaminhar páginas para serem analisadas em um servidor Web. Apesar da ferramenta Fujitsu oferecer a verificação de conformidade tanto com as WCAG 1.0 quanto com as *guidelines* próprias da Fujitsu, este fato causou alguma dificuldade nos participantes, que se confundiram novamente com as numerações diferentes para *checkpoints* equivalentes nos dois conjuntos de *guidelines*, pois mesmo quando se selecionam as WCAG 1.0, nas explicações contidas no relatório aparecem referências aos números das *guidelines* da Fujitsu.

Com relação aos checkpoints das WCAG 1.0 em si, durante a avaliação com a Fujitsu os participantes também tiveram algumas dúvidas e dificuldades. Um deles não soube informar com precisão se os equivalentes textuais fornecidos são efetivamente "significativos", conforme recomendação do *checkpoint* 1.1. Outra falha verificada pelo mesmo participante foi com relação ao *checkpoint* 10.1, que recomenda não abrir novas janelas do navegador sem avisar o usuário: o texto contido na Fujitsu é ligeiramente diferente daquele contido nas WCAG 1.0. Ainda, os outros dois participantes ficaram em dúvida se os *checkpoints* 1.1 e 6.5 são equivalentes no caso de *applets*, já que o texto explicativo mostrado pela ferramenta é o mesmo para ambos (o atributo "alt" está declarado para o *applet*). De fato, e segundo as próprias WCAG 1.0, alguém pode interpretar que, ao incluir um "alt" para um *applet*, ao mesmo tempo se está atendendo o *checkpoint* 1.1, de oferecer um equivalente textual para todo elemento não textual, quanto o 6.5, que diz que o conteúdo dinâmico deve ser acessível ou deve haver uma apresentação

alternativa, ou seja, um “alt” – um dos exemplos nas WCAG 1.0 de técnicas para satisfazer o *checkpoint* 6.5 é exatamente esta!

A análise das respostas obtidas nos questionários respondidos pelos participantes mostrou uma preferência pelas ferramentas Fujitsu e Truwex quanto à apresentação dos resultados da verificação de conformidade com as WCAG 1.0 e aos relatórios apresentados pelas mesmas. Também, todos os participantes concordaram que a experiência com as WCAG 1.0 e as ferramentas de verificação de *checkpoints* contribuiu para melhorar o entendimento dos mesmos sobre acessibilidade. No entanto, dois dos três participantes consideraram inadequado o tempo necessário para se avaliar uma página Web, independente da ferramenta usada. Além disto, um dos participantes comentou sobre a subjetividade de alguns *checkpoints* e *guidelines* das WCAG 1.0, como a *guideline* 14 (garanta que os documentos são claros e simples) e também da necessidade do avaliador de acessibilidade em conhecer bem as WCAG 1.0 especialmente para realizar a verificação manual dos *checkpoints*.

Uma constatação relevante e que merece destaque é que, apesar da experiência e do aprendizado a respeito das WCAG 1.0 adquiridos ao longo dos testes com as quatro ferramentas, todos os participantes, sem exceção, tiveram dificuldades em lidar com as WCAG 1.0 do início ao final dos testes e precisaram, em média, de uma hora para avaliar uma única página Web, independente da ferramenta. Para se ter noção, tipicamente, são necessárias duas horas para avaliar a usabilidade de um sistema como um todo com as heurísticas de Nielsen – no mesmo período seria possível avaliar a conformidade com as WCAG 1.0 de apenas duas páginas Web.

Em resumo, pode-se dizer que as ferramentas auxiliam a verificação de conformidade com as WCAG 1.0, mas não fazem milagres a ponto de minimizar a sua complexidade ou mesmo resolver as falhas presentes e já descritas anteriormente. Os resultados obtidos deste estudo com as ferramentas de verificação das WCAG 1.0 foram submetidos para um periódico internacional da área e espera-se que sejam publicados em breve.

3.6.4. Testes com usuários

Para avaliar acessibilidade, também é possível a realização de testes com usuários, assim como se faz na avaliação de usabilidade. Inclusive, as mesmas técnicas e procedimentos apontados no capítulo anterior são aplicáveis a testes com usuários com deficiência, porém com algumas adaptações. Por exemplo, não se imagina aplicar um questionário impresso para avaliar a satisfação subjetiva de um usuário cego. Mais do que quando se avalia usabilidade, em testes com usuários com deficiências é importante entender os atributos dos usuários e o avaliador deveria ganhar a confiança dos participantes antes de iniciar a avaliação (Lepistö e Ovaska, 2004).

Adicionalmente, de acordo com Lepistö e Ovaska (2004), testes convencionais tendem a não funcionar com usuários com deficiências cognitivas e/ou dificuldades permanentes relacionadas à cognição. Primeiro, porque as possíveis dificuldades em ler atrapalham a compreensão da lista de tarefas, que comumente é entregue aos usuários na forma impressa. Segundo, o experimentador precisa ser mais ativo que o usual em um teste com usuários deste tipo caso adote uma lista de tarefas impressa. Terceiro, é fundamental que os usuários não tenham qualquer pressão de tempo para o teste. Finalmente, questionários também deveriam ser evitados pelos mesmos motivos.

Sendo assim, Lepistö e Ovaska (2004) sugerem que, para testes com usuários com deficiências cognitivas, seja aplicado um "percurso informal". Ou seja, não é dada uma lista de tarefas aos usuários participantes, mas sim deixado que cada participante explore a interface à sua maneira. Foi desta forma, inclusive, que foram realizados os testes de acessibilidade do HagáQuê com crianças com diferentes deficiências e dificuldades cognitivas (Tanaka, 2005): simplesmente pedia-se para que a criança tentasse fazer uma história em quadrinhos no HagáQuê, usando quaisquer recursos oferecidos pelo software, sem qualquer limite de tempo pré-estabelecido. Também, Lepistö e Ovaska (2004) recomendam entrevistas para coletar medidas qualitativas, com questões simples e claras e sempre esperando as respostas do participante, sem tentar antecipá-las.

Para outros tipos de usuários, outras adaptações nos testes são necessárias. No caso de surdos sem um domínio efetivo da língua escrita e falada, mas fluente na língua de

sinais, por exemplo, muitas vezes os testes devem ser realizados com a presença de um intérprete de língua de sinais, como em Tanaka et al. (2005).

Outro problema apontado por Bailey e Burd (2006) é com relação ao recrutamento de potenciais usuários para os testes. Segundo Bailey e Burd, encontrar usuários com diferentes deficiências para participar dos testes pode ser uma tarefa difícil. E mais: o número de usuários para testes para avaliar acessibilidade deveria ser maior do que para usabilidade, pois é preciso garantir uma cobertura razoável de deficiências, habilidades e necessidades. Bailey e Burd também destacam que testes com usuários com deficiências tendem a ser caros e demorados (Bailey e Burd, 2006), o que também constatamos ao realizar as avaliações de acessibilidade do HagáQuê.

Ainda que haja dificuldades inerentes ao se realizar testes com usuários com deficiência, os mesmos nunca deveriam ser descartados. O ideal é que sejam aplicados de forma complementar a outros métodos de inspeção de acessibilidade, assim como é recomendado para avaliar usabilidade.

3.6.5. Percurso de barreiras

Brajnik (2006), insatisfeito com avaliações de conformidade com *guidelines*, propôs o percurso de barreiras (ou *Barriers Walkthrough*) como um método de inspeção de interfaces Web para se encontrar problemas de acessibilidade. Segundo Brajnik, uma barreira é uma condição que dificulta as pessoas de atingirem um objetivo quando navegam em um site, especialmente quando auxiliadas por tecnologias assistivas. Um exemplo de barreira pode ser visto na tabela 3.4. O conjunto completo de barreiras também foi disponibilizado por Brajnik (2009).

Tabela 3. 4. Exemplo de barreira segundo Brajnik (2006).

Barreira	Usuários não podem identificar ou entender a informação contida em uma imagem (exemplos: um diagrama, um histograma).
Falha	Uma imagem que não possui um texto associado (como um atributo ALT, texto corrido próximo à imagem ou como link para uma página separada.
Usuários afetados	Usuários cegos que usam leitores de tela
Consequências	Usuários procurarão por mais explicações; gastarão tempo e esforço substanciais; eficiência, produtividade e satisfação são severamente afetadas.

Para aplicar o percurso de barreiras de Brajnik, é preciso identificar cenários, que são compostos por tipos de usuários, configurações, objetivos e possíveis tarefas. Os tipos de usuários sugeridos por Brajnik são: usuários cegos com leitores de tela, usuários com visão subnormal com lentes de aumento, usuários com deficiências motoras com um teclado normal e/ou mouse apenas, usuários surdos e usuários com deficiências cognitivas (com dificuldades de escrita e leitura e/ou com déficit de atenção). Para cada tipo de usuário, há um conjunto de potenciais barreiras a serem verificadas se existem nas páginas Web (Brajnik, 2009).

Os objetivos e tarefas dos usuários variam de acordo com a aplicação, e Brajnik inclusive recomenda extraí-los a partir de documentos de casos de uso. Desta forma, cada tarefa/objetivo terá um conjunto de diferentes páginas associadas (ou percurso de páginas). Então, o avaliador deve averiguar se, dado o tipo de usuário e suas tarefas/objetivos, há barreiras nas páginas associadas.

Ao final, como em uma avaliação heurística de usabilidade, os avaliadores se reúnem para produzir uma lista dos problemas encontrados, associando um grau de severidade para cada um dos problemas.

3.6.6. Percurso cognitivo estendido

Além do percurso de barreiras de Brajnik, Kato e Hori (2006) propuseram um novo percurso cognitivo para identificar problemas de usabilidade e acessibilidade.

O percurso cognitivo estendido de Kato e Hori procura realizar a distinção entre objetos e ações e entre percepção e entendimento, uma vez que apenas perceber uma informação, visível ou audível, não garante que a mesma seja entendida cognitivamente. Para tal, as quatro questões típicas do percurso cognitivo tradicional foram alteradas, chegando-se a um conjunto de nove questões a serem respondidas pelo avaliador na fase de análise, quando deve percorrer o conjunto de ações para completar uma tarefa. As novas questões, no entanto, não foram explicitamente publicadas por Kato e Hori até o momento de escrita desta tese.

Ainda assim, na pesquisa de Kato e Hori (2006), notou-se que o percurso cognitivo estendido tende a encontrar mais problemas na interface do que a versão tradicional, embora a fase de análise se torne mais demorada.

3.6.7. Heurísticas de acessibilidade já existentes

Hoje em dia, já existem heurísticas de acessibilidade, mas nenhuma delas atende aos objetivos desta tese de doutorado.

Paddison e Englefield (2003), especialistas em usabilidade da IBM, desenvolveram um conjunto de heurísticas de acessibilidade para a Web baseadas em pesquisa de documentos de *guidelines*, entre os quais se considerou as WCAG 1.0 (Chisolm et al., 1999) e a Section 508 (Section 508, 2000). Embora o conjunto de heurísticas de Paddison e Englefield possa ser considerado uma abstração das *guidelines* existentes, ainda é muito centrado nas *guidelines* de acessibilidade na Web que serviram de base, o que dificulta seu uso para a avaliação em sistemas não-Web. A lista de suas heurísticas pode ser vista a seguir.

1. Fornecer alternativas significativas e relevantes para elementos não-textuais
2. Suportar navegação por *tags* consistente e correta
3. Permitir uso completo e eficiente do teclado
4. Respeitar configurações do navegador do usuário
5. Garantir uso adequado de padrões e controles proprietários
6. Não usar apenas cores para distinção de informação
7. Permitir aos usuários controle sobre potenciais distrações
8. Permitir aos usuários entender e controlar limites de tempo
9. Garantir que o conteúdo do Web site seja compatível com tecnologias assistivas

Koivunen e McCathieNevile (2001) também desenvolveram um conjunto de heurísticas de acessibilidade para a Web, porém destinados à acessibilidade de gráficos e elementos multimídia. O conjunto resultante foi baseado em pesquisas de *guidelines* da W3C para conteúdo Web, incluindo as WCAG 1.0, as *User Agent Accessibility Guidelines* 1.0 (UAAG 1.0) (Jacobs et al., 2002) e o *feedback* que Koivunen e McCathieNevile receberam quando trabalharam em outros documentos de acessibilidade para SMIL, SVG e XML. As heurísticas propostas por Koivunen e McCathieNevile podem ser vistas a seguir.

1. Ofecerer alternativas equivalentes
2. Ofecerer maneiras de selecionar conteúdo equivalente
3. Oferecer ao usuário controle para apresentação
4. Oferecer interação independente de dispositivo
5. Oferecer semântica para estruturas
6. Oferecer componentes reusáveis

As heurísticas de Koivunen e McCathieNevile não foram usadas para avaliar quaisquer interfaces e são, na verdade, uma forma diferente de descrever o conteúdo de vários documentos da W3C a respeito de acessibilidade em gráficos e multimídia.

Outra publicação a respeito de heurísticas de acessibilidade é de Wattenberg (2006). Para Wattenberg, aprender a usar um computador é mais difícil para os usuários que precisam de leitores de tela e, dentre os usuários de tecnologias assistivas, estes são os que

mais crescem – e, de fato, os dados de censos demográficos como os do IBGE (2000) comprovam que há mais pessoas com deficiências visuais do que com outros tipos de deficiências. Sendo assim, propôs levantar um conjunto de heurísticas específicas que possibilitasse identificar problemas de usabilidade e de acessibilidade em leitores de telas. Porém, até o momento em que esta tese é escrita, os resultados do trabalho de Wattenberg ainda não foram divulgados[15]. Detalhes de como as heurísticas foram ou serão extraídas também não estão disponíveis.

15 Segundo Wattenberg (2006), as heurísticas serão publicadas em uma página de seu site, que continua indisponível até o presente momento (29/dez/2009): http://www.wattenberg.biz/Alt_Learning_Project.htm.

Capítulo 4

Processo de elaboração das heurísticas

Como visto no Capítulo 1, não existe um único método para se obter heurísticas. Ao contrário, diferentes métodos podem ser aplicados (baseados em conhecimento popular, em experiência, em avaliações, em pesquisa, ou em assertivas de design baseado em cenários).

No caso desta tese, pode-se dizer que o processo de elaboração do conjunto de heurísticas de acessibilidade se desenvolveu através de duas grandes fases e é diferente dos demais métodos vistos anteriormente.

A primeira fase, de extração das heurísticas, contou com a colaboração de outro aluno de pós-graduação do Instituto de Computação da UNICAMP e que também era especialista em acessibilidade. O objetivo era que nós explorássemos de maneira informal algumas interfaces em busca de problemas de acessibilidade. Durante cada exploração, ambos realizavam anotações sobre os problemas encontrados, justificando os problemas baseados nas suas experiências na área de acessibilidade e design universal, a fim de facilitar a extração de heurísticas deste processo. Ou seja, pode-se dizer que esta fase se assemelha um pouco aos métodos de obtenção de heurísticas baseados em avaliações e em experiências de especialistas. Não foi selecionado um método baseado em pesquisa, pois as principais fontes dos mesmos são *guidelines* e *checkpoints* de acessibilidade para a Web que, inclusive, já serviram de base para outras heurísticas de acessibilidade (Paddison e Englefield, 2003; Koivunen e McCathieNevile, 2001).

Na segunda fase, de refinamento das heurísticas de acessibilidade, foram realizados dois experimentos a fim de validar o conjunto de heurísticas proposto e potencialmente melhorar a descrição das mesmas. Nesta fase, foram recrutados alguns participantes especialistas em avaliação de usabilidade para avaliar a acessibilidade de uma interface utilizando as heurísticas propostas. A adoção desta fase de avaliação e refinamento no

processo de obtenção de heurísticas nem sempre é efetuada por outros pesquisadores que propuseram heurísticas, como nos casos de Koivunen e McCathieNevile (2001) e Somervell et al. (2003), embora muitos outros trabalhos tenham adotado ao menos algum tipo de avaliação do conjunto de heurísticas para validá-lo (Cuperschmid, 2008; Pinelle et al., 2008; Paddison e Englefield, 2003).

A figura 4.1 ilustra o modelo adotado para a elaboração das heurísticas, no qual os passos em amarelo representam a fase de extração, os passos em verde a fase de refinamento e o passo final, em azul, a publicação da versão final das heurísticas. Nas próximas subseções, cada uma das fases de elaboração das heurísticas será descrita com maiores detalhes.

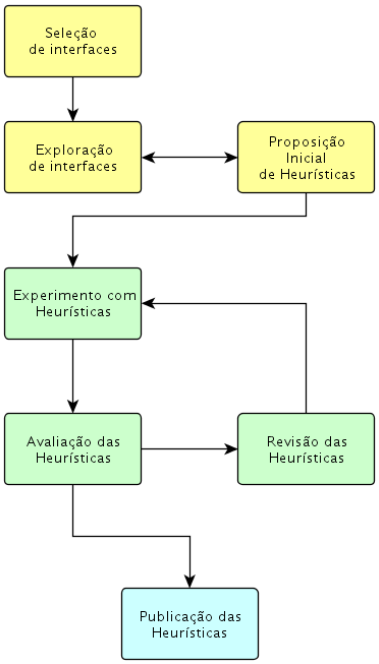


Figura 4. 1. Processo de elaboração das heurísticas de acessibilidade.

4.1. Fase de extração

Na fase de extração das heurísticas, o objetivo era encontrar um conjunto inicial de heurísticas de acessibilidade, capazes de garantir uma boa cobertura de problemas. Também, deveria ser possível chegar a um conjunto de heurísticas que não resultassem em "falsos positivos", ou seja, que não fossem identificados problemas que não são verdadeiros.

Durante esta fase, eu e outro experiente especialista em acessibilidade exploramos nove interfaces, procurando por potenciais problemas de acessibilidade. E, quando eu dizia "isto é um problema de acessibilidade", o outro especialista deveria me questionar, para entender o porquê considerar aquilo como um problema, fazendo com que expressasse o conhecimento muitas vezes informal que eu possuía sobre problemas de acessibilidade. Adicionalmente, tanto eu quanto o outro especialista realizamos anotações sobre as razões de se considerar isto ou aquilo um problema de acessibilidade. Ou seja, estas inspeções serviam essencialmente para externalizar o meu conhecimento a respeito de problemas de acessibilidade.

As interfaces escolhidas para inspeção foram de software livres e/ou gratuitos, a maioria educacionais, que poderiam ser obtidos com facilidade, e que ofereciam diferentes meios de interação: escrita, clique, arrastar e soltar, selecionar, dentre outros. As interfaces avaliadas nesta fase e os problemas encontrados podem ser vistos no Anexo B, juntamente com alguns dos comentários feitos durante as avaliações.

Como resultado do processo de exploração, obteve-se um conjunto manuscrito de problemas de acessibilidade, de questionamentos, justificativas e comentários a respeito dos mesmos. A partir disto, procurei abstrair os problemas encontrados na forma de um conjunto reduzido de heurísticas. E para cada heurística foi adicionado um breve texto explicativo. Assim, foi concebida a primeira versão das heurísticas de acessibilidade:

1. **Suporte a diferentes tipos de entrada e saída**

   O sistema deve ser utilizável através dos dispositivos de entrada e saída que o usuário adotar, não importa quais sejam. Em especial, o sistema deve suportar a interação por meio de teclado, mouse e leitor de telas.

2. **Conteúdo para todos os usuários**

   Todo conteúdo presente no sistema deve ser compreensível para todo tipo de usuário, o que pode incluir muitas vezes fornecê-lo de diferentes formas. Alguns exemplos incluem:

   Figuras: uma descrição textual da mesma (para cegos utilizadores de leitores de tela)

   Músicas e Podcasts: letra ou transcrição (para surdos oralizados)

   Texto escrito: língua de sinais, sinalizada na forma de animação, escrita de sinais ou substituir informação textual por ícones, animações ou outros componentes visuais (para surdos não oralizados)

   Diferenciação de informação por meio de cor, como campos obrigatórios de um formulário: marcação também com uso de símbolos (para pessoas com cromodeficiências).

3. **Independência de uso**

   Usuários com deficiências devem ser capazes de ligar, desligar e configurar todas as funcionalidades relacionadas à acessibilidade que o sistema fornece de forma autônoma, sem a ajuda de terceiros. Tais funcionalidades podem ser aumentar o tamanho da fonte, habilitar o teclado virtual ou a lupa/lente de aumento, alternar para texto com alto contraste de cores, reduzir ou aumentar o tempo de varredura do teclado virtual, entre outros.

4. **Respeito às preferências do usuário**

   Usuários possuem diferentes particularidades e habilidades e, desta forma, podem necessitar de configurações específicas para as cores usadas na tela, o volume dos alto-falantes, o tamanho e o tipo de fonte dos textos, o tempo de resposta, entre outros. Portanto, o sistema deve respeitar tais configurações definidas pelo usuário,

principalmente se tais configurações foram definidas de forma global (no sistema operacional ou no navegador de Internet, por exemplo).

5. **Eficiência em navegação alternativa**

   Um sistema pode se mostrar muito ineficiente para os utilizadores de leitores de telas, teclados virtuais e outras tecnologias assistivas. Por exemplo, ao navegar por um site que não conhecem, muitas vezes os usuários de leitores de telas são obrigados a ler muitos parágrafos antes de chegar ao que efetivamente importa ou desejam. Assim, o sistema deve oferecer teclas de atalho para facilitar o acesso às principais funcionalidades e, no caso de conteúdos textuais, deve procurar agrupá-los de alguma forma, incluir meios de acessar rapidamente estes grupos e informar apenas o necessário por meio de um diálogo simples e direto.

Para exemplificar a aplicação das heurísticas, alguns problemas encontrados na fase de extração podem ser associados às mesmas, como visto na listagem a seguir. Uma descrição mais detalhada de cada software está disponível no Anexo B.

- **Software:** Quebra-Cabeça. **Problema:** O jogo do Quebra-Cabeça só funciona com o mouse, através da ação de arrastar e soltar, sendo impossível o uso do mesmo apenas com o teclado – o que impede seu uso por pessoas com deficiências motoras, por exemplo. Também, seria praticamente impossível usá-lo com leitores de tela, uma vez que, em geral, não se usa o mouse ou outros dispositivos de apontamento com tais tecnologias assistivas. **Heurística(s) violada(s):** Suporte a diferentes tipos de entrada e saída.
- **Software:** Jogo da Reciclagem. **Problema:** As lixeiras são diferenciadas apenas através da cor, fazendo com que a identificação de cada uma possa ser difícil por pessoas com cromodeficiências ou monitores monocromáticos. Na vida real, sempre há rótulos em lixeiras de coleta de lixo reciclável, mas no jogo não. **Heurística(s) violada(s):** Conteúdo para todos os usuários.

- **Software:** Cada Caso é um Caso. **Problema:** O software apresenta elementos escondidos: textos em páginas Web que têm a mesma cor do fundo para que fiquem "invisíveis" para usuários videntes. Porém, o uso de tais elementos deve ser feito com cuidado, uma vez que eles podem ser lidos por leitores de telas. Ou seja, no mínimo, incluir este tipo de elemento irá aumentar a quantidade de texto a ser lido pelo leitor de telas e potencialmente reduzir a eficiência de uso da interface com este tipo de tecnologia assistiva. **Heurística(s) violada(s):** Eficiência em navegação alternativa.
- **Software:** Geometria. **Problema:** O software Geometria ignora as configurações de cores definidas no sistema operacional. Assim, usuários que precisam usar esquemas de cores em alto contraste poderão sentir dificuldades ao interagir com o software. **Heurística(s) violada(s):** Respeito às preferências do usuário.

4.2. Fase de refinamento

Terminada a extração inicial das heurísticas, iniciou-se a fase de refinamento, cujo principal objetivo era realizar a avaliação das heurísticas em si, colocando-as em uso e coletando opiniões de avaliadores de interfaces. Adicionalmente, dependendo do resultado da avaliação, as heurísticas deveriam ser revisadas – o que ocorreu de fato. Esta fase pode ser considerada um ciclo iterativo que deve ser repetido até que se chegue a um conjunto de heurísticas que atendam alguns critérios pré-definidos.

Somervell e McCrickard (2005), autores de um conjunto de heurísticas de usabilidade específicas para sistemas de exibição de informações em telas grandes, citam alguns possíveis critérios ou métricas a serem aplicados para se avaliar novos conjuntos de heurísticas, dentre os quais se destacam a aplicabilidade e a completude[16]. A aplicabilidade é a possibilidade de se associar ao menos uma das novas heurísticas propostas aos problemas existentes na interface ao passo que a completude é a razão entre o número de

[16] Os demais critérios descritos no trabalho de Somervell e McCrickard (2005) são basicamente derivados da aplicabilidade e da completude.

problemas encontrados com o conjunto de heurísticas e o total de problemas existentes em uma interface (o cálculo do total de problemas existentes e exige a aplicação de, pelo menos, mais um método de avaliação de interface). No caso específico desta tese de doutorado, foram estabelecidos os seguintes critérios:

1. Aplicabilidade: os avaliadores deveriam conseguir relacionar os problemas de acessibilidade encontrados com uma ou mais heurísticas.
2. Satisfação dos avaliadores: os avaliadores deveriam ficar satisfeitos com os resultados obtidos após uma avaliação de acessibilidade com as heurísticas.
3. Tempo necessário para aprender e aplicar as heurísticas: o tempo necessário para avaliar uma interface impacta diretamente no custo final do sistema. O ideal seria que a avaliação de acessibilidade com as heurísticas propostas fosse tão rápida quanto com as heurísticas de usabilidade de Nielsen – tipicamente, uma sessão de avaliação individual usando as heurísticas de Nielsen leva duas horas para um sistema inteiro (Rocha e Baranauskas, 2003), ao passo que uma verificação de conformidade com as WCAG 1.0 demanda em média uma hora para uma única página Web, independente da ferramenta semi-automática adotada, conforme resultados do estudo descrito no Capítulo 3.

Por ser um cálculo muito mais custoso, que exige a descoberta de **todos** os problemas de acessibilidade de uma interface, a completude não foi considerada no momento de avaliar as heurísticas de acessibilidade propostas.

Dois ciclos da fase de refinamento foram realizados e, em cada um deles, quatro avaliadores de interfaces foram recrutados para participar de uma avaliação de acessibilidade usando as heurísticas propostas, de forma bem similar ao processo de avaliação heurística de usabilidade de Nielsen (1994). Este processo de avaliação com as heurísticas de acessibilidade propostas foi chamado informalmente de experimento na comunicação com os participantes.

Assim, nestes experimentos, primeiramente os participantes avaliaram individualmente a interface, explorando-a e tentando relacionar os problemas encontrados com uma ou mais heurísticas. Finalizada a avaliação individual, todos os participantes se

reuniram para discutir as descobertas de cada um e chegar a uma lista consolidada de todos os problemas encontrados.

A diferença da avaliação heurística de usabilidade de Nielsen para a destes experimentos está exatamente no fato das heurísticas em si também estarem sendo avaliadas. Para isto, acompanhei inteiramente cada etapa de avaliação de acessibilidade nos dois experimentos executados, desde as avaliações individuais até a reunião final entre os participantes avaliadores, para que fosse possível obter uma percepção das dificuldades no uso das heurísticas e a opinião dos participantes a respeito das mesmas. Também foram efetuadas algumas medições, como o número de problemas encontrados e o tempo necessário para se fazer a avaliação de acessibilidade com as heurísticas.

4.2.1. Primeiro Experimento

O primeiro experimento, ou primeiro ciclo da fase de refinamento, teve como entrada as heurísticas obtidas na fase de extração. O processo de avaliação das heurísticas para refinamento será descrito a seguir, bem como os resultados obtidos.

4.2.1.1. Materiais e métodos

Foram selecionados quatro participantes para o primeiro experimento, todos especialistas em avaliação de usabilidade e alunos de pós-graduação em Ciência da Computação na UNICAMP. Um dos participantes é mestrando e especialista em acessibilidade, especialmente para surdos, enquanto os outros três são doutorandos. Os quatro já tinham algum conhecimento sobre as WCAG 1.0, embora apenas o especialista em acessibilidade usasse as *guidelines* com freqüência.

Foi apresentado aos participantes selecionados um termo de consentimento, cuja finalidade era esclarecer os objetivos do experimento em questão, bem como garantir sua

privacidade quando da publicação de resultados do experimento e assegurar que poderiam desistir a qualquer momento de participar do mesmo. Uma cópia do termo de consentimento pode ser encontrada no Anexo C.

Como alguns dos participantes não trabalhavam frequentemente com acessibilidade e design universal, todos foram reunidos e foi realizada uma breve apresentação para elucidar a importância da acessibilidade e que algumas pessoas poderiam acessar um sistema de informação de diferentes formas e de diferentes dispositivos, de acordo com as suas habilidades e preferências. Na mesma apresentação, foi também abordado como normalmente se avalia acessibilidade (verificação de conformidade com *guidelines* e testes com usuários) e os problemas e dificuldades inerentes a estas avaliações. Desta forma, era esperado que todos os participantes estivessem cientes e sensibilizados sobre o contexto geral que envolve uma avaliação de acessibilidade e a importância da mesma no desenvolvimento de qualquer software.

Em seguida, cada uma das cinco heurísticas iniciais, extraídas da fase de exploração, foi apresentada: o seu título, a sua descrição e um exemplo de violação. Paralelamente, os participantes receberam uma versão impressa das heurísticas e outro documento impresso contendo exemplos de violações, que pode ser visto no Anexo C. À medida que cada uma das heurísticas era apresentada, os participantes tinham a oportunidade de questioná-las e analisar como poderiam aplicá-las, que outras interfaces potencialmente as violariam e manifestar seu grau de entendimento sobre o título e a descrição de cada uma delas. A apresentação do tema acessibilidade e das heurísticas em si durou aproximadamente 2 horas e meia, tendo em vista que os participantes já eram avaliadores de usabilidade e não totalmente leigos em acessibilidade, não necessitando de maiores aprofundamentos. Caso fossem recrutados participantes inexperientes, certamente haveria a necessidade de um treinamento mais abrangente, envolvendo conceitos básicos de acessibilidade – no caso, e baseado na minha experiência em lecionar sobre acessibilidade e avaliação de acessibilidade, estimo que entre 4 a 6 horas seja mais do que suficiente para realizar um treinamento deste tipo.

Depois de retiradas as dúvidas iniciais sobre as heurísticas, a próxima etapa do experimento envolveu a aplicação das heurísticas, seguindo o mesmo processo consagrado na literatura (Nielsen, 1994). Como todos os participantes já eram experientes avaliadores

de usabilidade, não foi necessária uma apresentação ou treinamento a respeito de avaliação heurística, apenas foi esclarecido que o processo de avaliação seria similar à avaliação heurística de usabilidade de Nielsen (1994): avaliações individuais seguidas por uma reunião para discutir os problemas encontrados.

Então, cada avaliador participante recebeu a tarefa de avaliar individualmente o editor de textos do Google Docs[17] durante, no máximo, 45 minutos, utilizando as heurísticas de acessibilidade propostas. Para tal, os participantes se dirigiram a um laboratório de informática e cada um sentou-se em frente a um computador com o sistema operacional Windows XP, o navegador Firefox 3 e o leitor de telas FireVox[18] instalado.

Como todos os avaliadores participantes realizaram as avaliações em paralelo e no mesmo local, pude permanecer sempre próximo para esclarecer as possíveis dúvidas dos participantes e anotar quaisquer dificuldades que tinham com a aplicação das heurísticas. Após a avaliação individual, os avaliadores participantes se reuniram comigo para compartilhar os problemas encontrados e quais heurísticas foram violadas para cada problema, durante cerca de uma hora. Assim, consegui identificar outras dificuldades e divergências relacionadas à interpretação das heurísticas.

Finalmente, depois da discussão dos problemas encontrados, os participantes foram convidados a preencher um questionário para avaliar seu nível de satisfação e entendimento das heurísticas. O modelo de questionário utilizado pode ser visto no Anexo C.

4.2.1.2. Resultados

Durante a apresentação das heurísticas, houve algumas dúvidas sobre como poderiam avaliar a acessibilidade usando as heurísticas propostas, sem conhecer especificamente quem são os usuários e suas características. Porém, assim como ocorre com a avaliação heurística de usabilidade, o avaliador não necessita conhecer exatamente quem são os usuários, mas assim mesmo consegue avaliar a usabilidade de uma interface

[17] http://docs.google.com/. Informação capturada em 20/jul/2009.
[18] http://firevox.clcworld.net/. Informação capturada em 20/jul/2009.

de forma satisfatória na maioria das vezes. Também é esperado que, com a realização de mais avaliações, o avaliador adquira experiência e saiba identificar melhor os problemas na interface, similar ao que ocorre com as heurísticas de usabilidade de Nielsen.

Foi esclarecido aos participantes que há, de fato, a necessidade de se fazer a inspeção da interface de forma ligeiramente diferente da inspeção de usabilidade. Idealmente, é essencial que a inspeção de acessibilidade seja feita com o uso de tecnologias assistivas, especialmente leitores de tela. Adicionalmente, é importante efetuar a inspeção de acessibilidade tentando-se ao máximo se colocar no lugar de outros usuários, por exemplo, inspecionando a interface usando apenas o teclado (sem o mouse) e ativando recursos de acessibilidade do sistema operacional e/ou do navegador de Internet: alterando as cores e tamanho das fontes, por exemplo. No caso de sites que oferecem recursos de acessibilidade, como aumentar o tamanho da letra ou alterar o contraste de cores usado, também é necessário ativar estes recursos e inspecionar a interface.

Após a apresentação das heurísticas, os participantes sentaram-se cada um na frente de um computador pessoal em um laboratório de informática para avaliar individualmente a acessibilidade do Google Docs utilizando as heurísticas, contando com a minha presença no caso da necessidade de esclarecimentos.

As dúvidas dos participantes ao avaliar o Google Docs com as heurísticas não estavam na interpretação das mesmas ou na atribuição delas aos problemas de acessibilidade encontrados, mas sim em como ativar e utilizar o leitor de telas FireVox, alterar o esquema de cores do Google Docs para refletir as preferências de cores em alto contraste definidas no sistema operacional, como aumentar e diminuir a fonte da interface e de como usar teclas de atalho no navegador e no próprio Google Docs para facilitar o uso com o teclado. Ou seja, as dificuldades encontradas durante a avaliação individual se devem à falta de experiência dos participantes em lidar com tecnologias assistivas, visto que apenas um deles estava familiarizado com elas.

Durante a reunião para apresentar os problemas encontrados, foi constatado que dois participantes tiveram dificuldades em diferenciar as heurísticas “Suporte a diferentes tipos de entrada e saída” e “Eficiência em navegação alternativa”, tanto que sempre indicavam violações de ambas as heurísticas para um mesmo problema de acessibilidade, o que gerou uma discussão com os demais participantes. Estas dúvidas, ao que tudo indica, se devem à

interpretação que fizeram dos títulos destas heurísticas. Um dos participantes afirmou que, de fato, ficou em dúvida com relação à eficiência do site se o leitor de telas não funciona com ele. Após relerem a descrição das mesmas, os dois participantes se convenceram de que as duas heurísticas tratam de problemas distintos: a primeira diz respeito à possibilidade de se utilizar o sistema com diferentes dispositivos de entrada e saída enquanto a segunda heurística analisa se a navegação é eficiente mesmo se utilizando tecnologias assistivas ou entrada e saída não convencionais. Outro participante, então, chegou a complementar que, para um problema de "Eficiência em navegação alternativa" ocorrer, é preciso que a tecnologia assistiva funcione com o sistema, ou seja, o sistema assegura o "Suporte a diferentes tipos de entrada e saída".

Os participantes sugeriram duas mudanças na descrição das heurísticas. Na heurística "Independência de uso", um participante sugeriu alterar "sem a ajuda de terceiros" por "sem a ajuda de outras pessoas", para evidenciar que tipo de ajuda a heurística está se referindo. Outra sugestão de outro participante foi substituir "compreensível" por "acessível" na heurística "Conteúdo para todos os usuários". Segundo o participante, compreensível é um termo muito forte, que envolveria garantir que o usuário sempre irá entender o conteúdo de tudo o que acessa. No entanto, termo semelhante é adotado por outros autores, inclusive nas páginas de acessibilidade da W3C, ao definir o que é acessível na Web: permitir que as pessoas com deficiência possam perceber, **entender**, navegar, interagir e contribuir com a Web (Henry, 2005).

Outro questionamento interessante surgiu com o fato da interface do Google Docs e os documentos criados nele serem todos em língua escrita (Português, Inglês, Espanhol, entre outras), não permitindo o uso de escrita de sinais ou sinalização de alguma forma, por exemplo, por meio de animação ou vídeo. Um dos participantes sugeriu então oferecer um serviço de tradução automática para a língua de sinais, pois, apesar de não ser totalmente eficaz, auxilia no entendimento do conteúdo e supera o problema de "Conteúdo para todos os usuários" que foi encontrado pelos participantes na avaliação. Tal tecnologia, de fato, já existe, como no serviço do Rybená (2009). Todavia, o participante especialista em acessibilidade para surdos comentou que, pela sua experiência, poucos surdos gostam deste tipo de tradução automática, pela ausência de um dicionário completo do Português para a

língua de sinais e dos típicos problemas que qualquer tradução automática está sujeita (por exemplo, tradução inadequada de palavras para o contexto em questão).

Ao final da reunião para discutir os problemas de acessibilidade encontrados, os participantes voltaram a citar algumas dificuldades que acreditavam que encontrariam ao realizar uma avaliação de acessibilidade com as heurísticas. Um deles considerou novamente difícil avaliar a acessibilidade porque o avaliador necessita saber sobre os usuários, e achou ser mais fácil se colocar no lugar do usuário ao avaliar usabilidade apenas. O mesmo participante sugeriu desligar o monitor ao fazer a inspeção da interface com o leitor de telas – ele havia sido o único a efetuar a inspeção desta forma. Outro participante reconheceu que a experiência do avaliador com acessibilidade influencia na facilidade ou na dificuldade em efetuar a avaliação, assim como é na avaliação heurística de usabilidade. Finalmente, um terceiro participante sugeriu novamente o uso de cenários e perfis para facilitar a avaliação.

De uma forma geral, os avaliadores encontraram 24 problemas distintos de acessibilidade com a avaliação heurística de acessibilidade, sendo a maior parte deles relacionados com a heurística "Suporte a diferentes tipos de entrada e saída". A Tabela 4.1 identifica a quantidade de problemas encontrados para cada uma das heurísticas e a Tabela 4.2 apresenta a quantidade de problemas encontrados por cada partipante. No Anexo D, há uma lista condensada descrevendo todos os problemas encontrados pelos participantes. Alguns exemplos de problemas encontrados também são descritos a seguir.

Tabela 4. 1. Problemas encontrados para cada uma das heurísticas no primeiro experimento.

Heurística	**Número de ocorrências**
Suporte a diferentes tipos de entrada e saída	9
Conteúdo para todos os usuários	2
Independência de uso	0
Respeito às preferências do usuário	7
Eficiência em navegação alternativa	7

- **Problema:** Os botões da barra de ferramentas não são visíveis quando o usuário configura o sistema para utilizar cores em alto contraste. **Heurística(s) violada(s):** Respeito às preferências do usuário. **Participantes que encontraram:** P1, P2, P3 e P4.
- **Problema:** Quando se insere uma tabela, o cursor vai para uma linha logo abaixo na tabela, e não para dentro da mesma. Alguém que só navega usando teclado teria que então voltar para a primeira célula da tabela usando as setas do teclado para começar a preenchê-la. **Heurística(s) violada(s):** Eficiência em navegação alternativa. **Participantes que encontraram:** P1.
- **Problema:** O leitor de telas não lê os botões de Salvar, Desfazer e Refazer. **Heurística(s) violada(s):** Suporte a diferentes tipos de entrada e saída. **Participantes que encontraram:** P2.

Tabela 4. 2. Problemas encontrados por participante no primeiro experimento.

Participante	Total de problemas encontrados	Encontrados também por outros participantes
P1	9	2
P2	10	4
P3	5	3
P4	8	4

A partir dos dados da Tabela 4.2, é possível calcular a média de problemas de acessibilidade encontrados por participante, que é exatamente 8 e equivalente a um terço do total de problemas encontrados. Adicionalmente, verificou-se uma baixa sobreposição entre os conjuntos individuais de problemas encontrados, conforme pode ser visto na coluna "Encontrados também por outros participantes" da Tabela 4.2 – por exemplo, dos 9 problemas encontrados por P1, apenas 2 foram encontrados também por outros participantes. Uma análise da lista de problemas encontrados disponível no Anexo D mostra que apenas um único problema de acessibilidade foi encontrado pelos quatro participantes – de fato, a grande maioria dos problemas (20, ao todo) foi descoberta por

apenas um dos participantes. Tais fatos são indícios que, de forma similar à avaliação heurística de usabilidade, diferentes avaliadores descobrem diferentes problemas e há a necessidade de se realizar inspeções de acessibilidade sempre por um grupo de avaliadores, nunca individualmente. Vale a pena destacar também que, como o Google Docs não oferece recursos de acessibilidade próprios tais como alterar o tamanho da fonte da interface ou o esquema de cores para alto contraste ou ainda um leitor de telas integrado, não houve violações da heurística "Independência de uso".

Com relação ao critério aplicabilidade, da lista de problemas, apenas um deles foi impossível mapear para ao menos uma heurística. Foi levantado por um participante que as teclas de atalho do Google Docs se confundem com as teclas de atalho do próprio navegador: ao clicar Ctrl+S para salvar, o usuário salva o documento aberto no Google Docs ou toda a página Web mostrada no navegador? Na discussão, chegou-se a propor a criação de uma nova heurística para abordar a questão de transparência de uso, no sentido de que funcionalidades relacionadas com acessibilidade não interfiram umas nas outras. E outra dúvida surgiu com a discussão: o que aconteceria caso o sistema operacional e/ou o navegador estivesse configurado para usar cores em alto contraste e o site em que se está navegando possua sua própria configuração de alto contraste? O usuário deveria ser impedido de usar as duas configurações ao mesmo tempo caso isto dificulte a leitura do conteúdo do site?

O grande ponto é que este tipo de conflito entre funcionalidades, especialmente os semelhantes à dúvida de alto contraste surgida na discussão com os participantes, é um tanto artificial e improvável de ocorrer com usuários reais, pois um usuário teria que deliberadamente ativar as duas opções de alto contraste. Se o site já estiver sendo mostrado de forma adequada com as configurações de alto contraste do sistema operacional e/ou do navegador, não há necessidade de o usuário ativar configurações semelhantes do site também. Pode-se até propor que configurações globais tenham precedência sobre configurações locais, mas isto poderia comprometer a liberdade do usuário em escolher qual configuração mais lhe agrada. E, neste ponto, parece muito mais sensato permitir ao próprio usuário ativar e desativar com facilidade as configurações que desejar, na hora em que desejar, como prevê a heurística "Independência de uso".

Já o conflito entre teclas de atalho encontrado por um dos participantes depende da compreensão da ordem de precedência dos atalhos e das funcionalidades. Pelo que se constatou, atalhos têm a seguinte ordem de precedência: Atalhos locais (aplicativo Web) – Atalhos do navegador – Atalhos do sistema operacional.

Com isto, se um usuário está no seu navegador Web e clica F1, abrirá a ajuda do navegador. Caso esteja fora do navegador, sem nenhum outro programa aberto, abrirá a ajuda do sistema operacional. Para usuários acessando páginas textuais Web simples e não aplicações na Web como o Google Docs, parece muito mais fácil entender esta diferença, pois o contexto de trabalho é bem visível: ou se está no navegador Web ou se está no sistema operacional. E é exatamente esta visibilidade de contexto que faz falta quando se está usando o Google Docs ou outros aplicativos baseados na Web, como editores de fotos, gerenciadores de emails, entre outros. Ou seja, não fica claro se um comando ou atalho será respondido pelo navegador Web ou se pela aplicação que está aberta nele. Este problema de visualização do contexto vai além da proposta desta tese de doutorado e das heurísticas de acessibilidade.

Os resultados dos questionários respondidos podem ser vistos na Tabela 4.3 e nos gráficos das Figuras 4.2 a 4.8, considerando a seguinte escala de respostas para as perguntas no questionário:

1. Discordo Totalmente
2. Discordo Parcialmente
3. Indiferente
4. Concordo Parcialmente
5. Concordo Totalmente

Tabela 4. 3. Resultados dos questionários do primeiro experimento (média, desvio padrão e moda).

Q	Questão	Média	Desvio Padrão	Moda
1	O tempo para realizar uma avaliação com as heurísticas de acessibilidade é satisfatório.	4,25	0,50	4
2	Aprendi rapidamente a usar as heurísticas.	4,25	0,50	4
3	Tive dúvidas iniciais sobre como usar as heurísticas.	3,50	1,73	4
4	As dúvidas que porventura eu tinha no início ao usar as heurísticas foram diminuindo com o progresso da avaliação e a(s) reunião(ões) para discutí-las.	4,50	0,58	4
5	As heurísticas de acessibilidade são mais fáceis de aplicar do que as *guidelines* da W3C.	4,25	0,96	5
6	O material de apoio cedido, contendo a descrição das heurísticas e exemplos de violações, é claro e de fácil compreensão.	4,50	0,58	4
7	Eu entendi do que trata a heurística "Suporte a diferentes tipos de entrada e saída".	4,25	1,50	5
8	Eu entendi do que trata a heurística "Conteúdo para todos os usuários".	4,75	0,50	5
9	Eu entendi do que trata a heurística "Independência de uso".	5,00	0,00	5
10	Eu entendi do que trata a heurística "Respeito às preferências do usuário".	5,00	0,00	5
11	Eu entendi do que trata a heurística "Eficiência em navegação alternativa".	3,25	1,50	2
12	Encontrei problemas graves de acessibilidade usando as heurísticas.	4,50	1,00	5
13	Encontrei problemas de acessibilidade que nem imaginava que existiam usando as heurísticas.	3,75	1,50	5
14	Tentaria usar novamente as heurísticas se fosse necessário avaliar a acessibilidade de outro sistema.	4,75	0,50	5
15	No geral, me senti satisfeito com os resultados da minha avaliação de acessibilidade.	4,50	0,58	4
16	Acredito que posso aprimorar mais minha capacidade de avaliar a acessibilidade usando as heurísticas outras vezes.	4,75	0,50	5

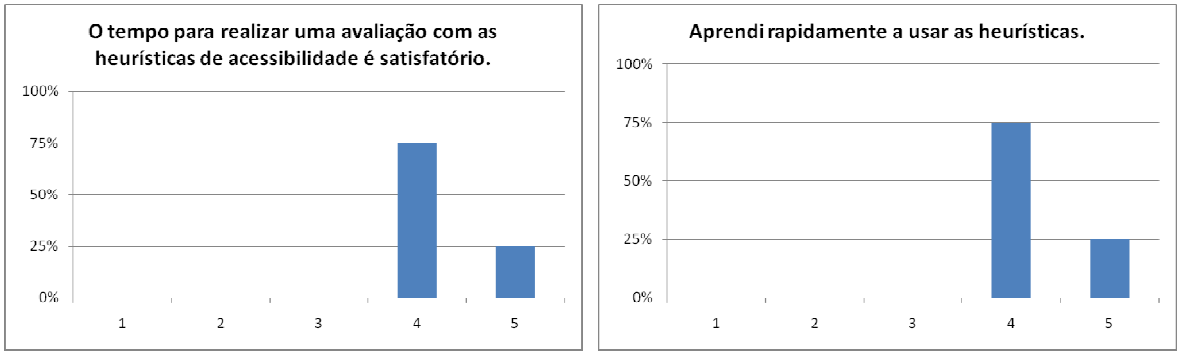


Figura 4. 2. Respostas dos avaliadores para as perguntas sobre os tempos dedicados a uma avaliação e para o aprendizado das heurísticas (primeiro experimento).

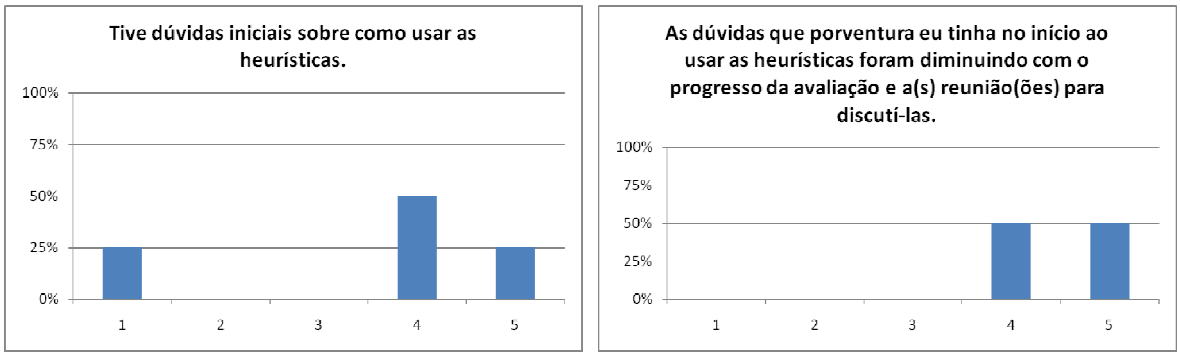


Figura 4. 3. Respostas dos avaliadores para as perguntas sobre dúvidas iniciais e a diminuição das mesmas com o avanço da avaliação (primeiro experimento).

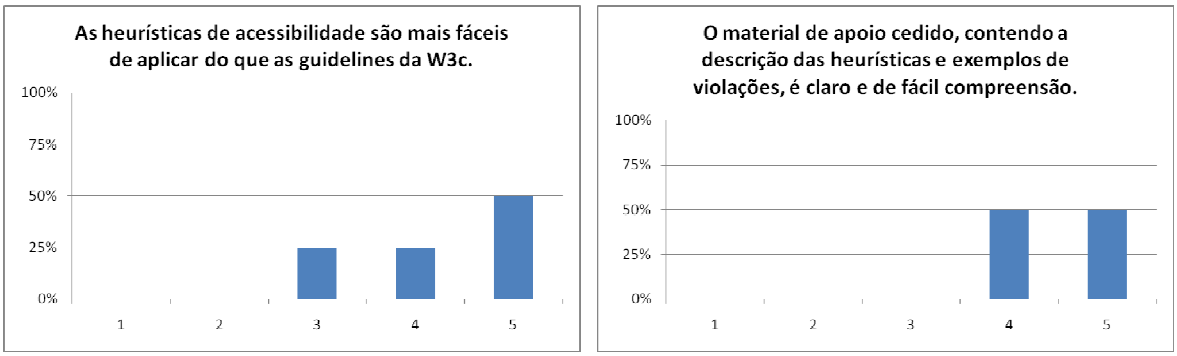


Figura 4. 4. Respostas dos avaliadores para as perguntas sobre a facilidade de aplicação das heurísticas comparadas às *guidelines* da W3C e sobre a facilidade do material de apoio cedido (primeiro experimento).

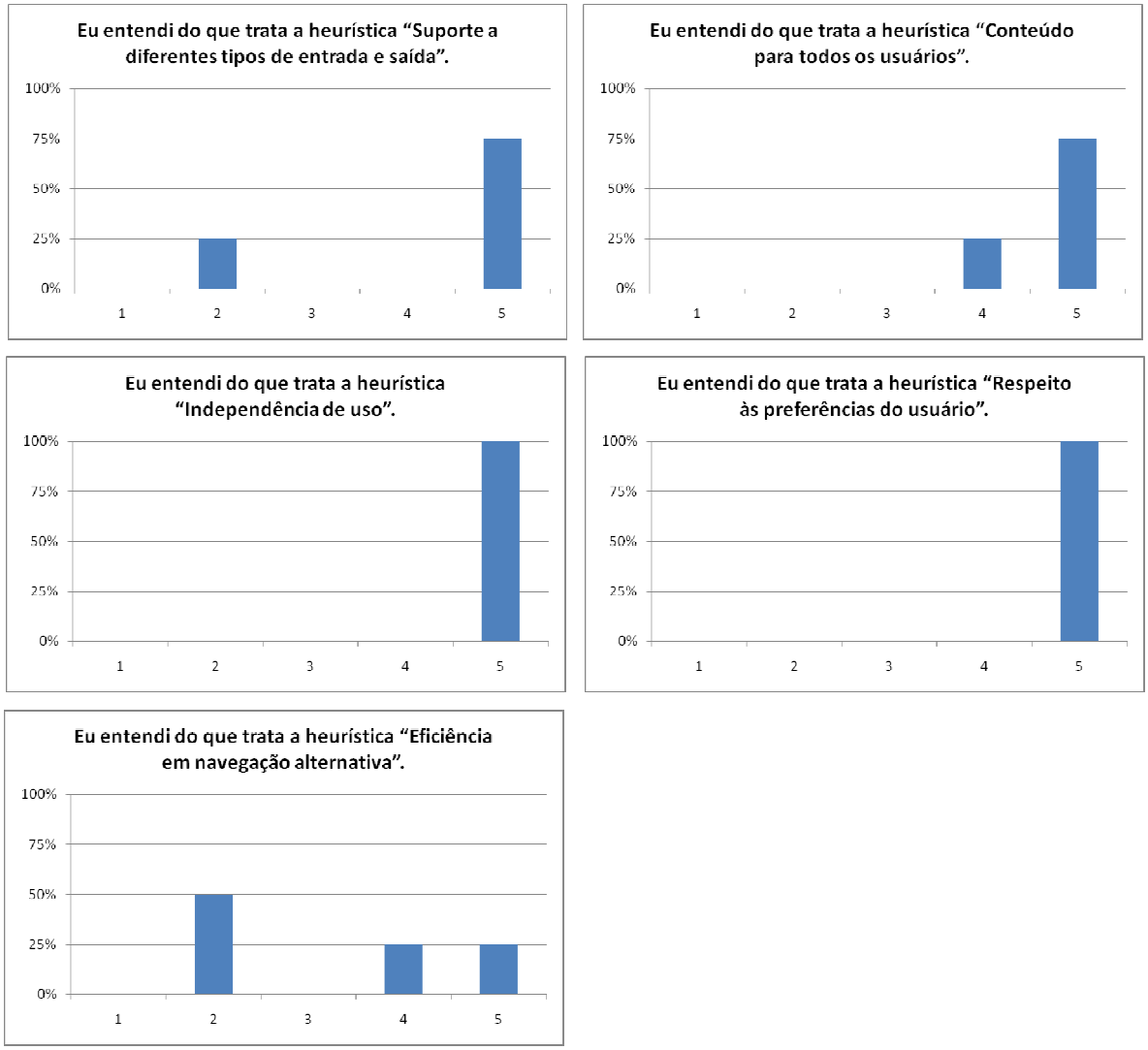


Figura 4. 5. Respostas dos avaliadores para as perguntas sobre o entendimento das heurísticas de acessibilidade (primeiro experimento).

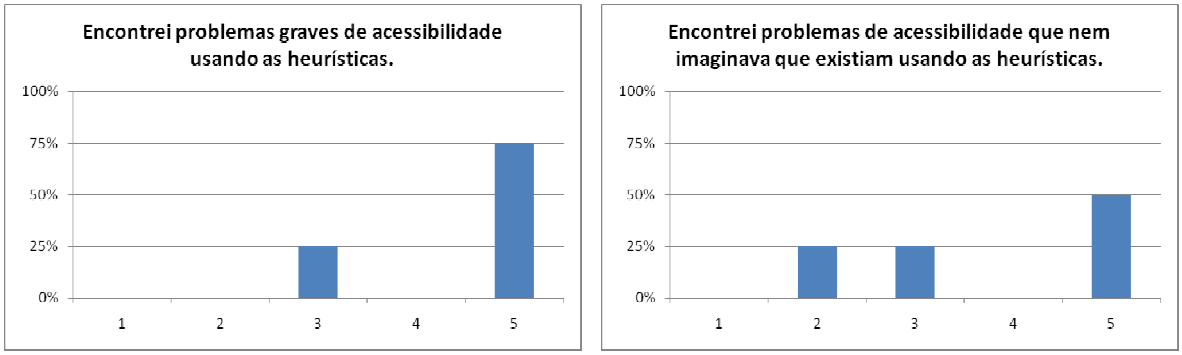


Figura 4. 6. Respostas dos avaliadores para as perguntas sobre os problemas encontrados (primeiro experimento).

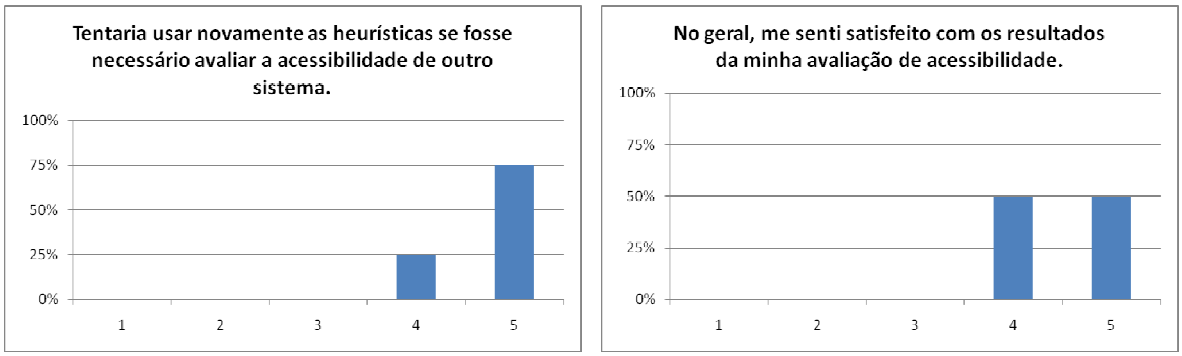


Figura 4. 7. Respostas dos avaliadores para as perguntas sobre a possibilidade de novo uso das heurísticas e a satisfação com os resultados obtidos (primeiro experimento).

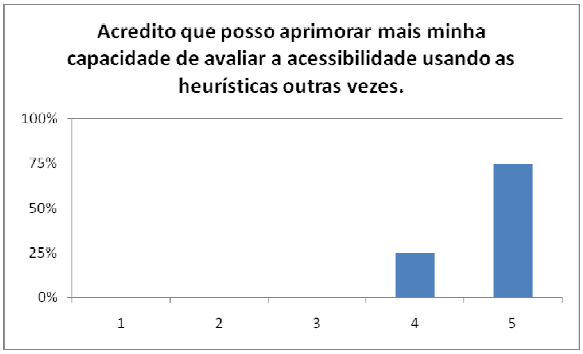


Figura 4. 8. Respostas dos avaliadores para as perguntas sobre a possibilidade de se melhorar as avaliações à medida que se torna mais experiente (primeiro experimento).

Como observado nos gráficos da Figura 4.3, 75% participantes (ou seja, três dos quatro) tiveram dúvidas iniciais no uso das heurísticas, mas todos, sem exceção, responderam que tais dúvidas foram diminuindo durante a avaliação e a reunião para discutir os problemas encontrados.

Ainda, foi possível notar resultados positivos de todos os participantes quando questionados sobre o tempo para realizar uma avaliação com as heurísticas de acessibilidade e comparando-as com as *guidelines* da W3C, conforme os gráficos das Figuras 4.2 e 4.4.

O material cedido para a avaliação de acessibilidade, contendo a descrição das heurísticas e exemplos de violações, também foi considerado adequado, conforme a Figura

4.4. Porém, como dois participantes sugeriram, poderia haver um conjunto de cenários, perfis ou *personas*[19] para facilitar a aplicação das heurísticas de acessibilidade.

Com relação às heurísticas em si, apenas a heurística "Eficiência em navegação alternativa" não foi satisfatoriamente entendida por todos os participantes, de acordo com os gráficos da Figura 4.5.

Finalmente, os participantes se sentiram satisfeitos com os resultados obtidos durante a avaliação com as heurísticas de acessibilidade e afirmaram que tentariam novamente utilizá-las caso necessitassem avaliar a acessibilidade de outros sistemas, conforme os gráficos da Figura 4.7.

Nos espaços destinados aos comentários dos usuários no final do questionário, repetiram-se parte das discussões ocorridas durante a reunião: dúvidas com relação às heurísticas "Suporte a diferentes tipos de entrada e saída" e "Eficiência em navegação alternativa" e sugestão para substituir "compreensível" por "acessível" na descrição da heurística "Conteúdo para todos os usuários".

Entre as opiniões sobre o que gostaram ao usar as heurísticas, os participantes citaram a facilidade de aplicá-las, a facilidade de entendê-las, o número reduzido de heurísticas, porém bastante abrangentes, os resultados muito satisfatórios obtidos na avaliação, a não necessidade de se ter conhecimentos de implementação (algo que, inevitavelmente, ocorre quando se faz uma verificação de conformidade com as WCAG 1.0) e a eficiência em se avaliar com as heurísticas propostas – usando as próprias palavras de um dos participantes, "avalia-se mais em menos tempo".

A respeito dos comentários do que não gostaram, os participantes listaram a dificuldade em se colocar no lugar dos usuários, a necessidade de uso de tecnologias assistivas, a dependência de contexto (sistema operacional e navegador Web) que ocorre ao avaliar com as heurísticas de acessibilidade e o tempo de preparação para avaliação (no caso, imaginar cenários e *personas* para auxiliar na avaliação). Um dos participantes sugeriu, ainda, uma descrição mais extensa em cada uma das heurísticas.

---

[19] O termo *persona* descreve um modelo de usuário e foca nos objetivos do usuário quando interagindo com um artefato. Diferente de um perfil de usuário, uma *persona* representa padrões de comportamento, os objetivos e as motivações de usuários, compilados em uma descrição fictícia de um indivíduo, que frequentemente inclui até mesmo um nome (João, Maria, etc.) e características de personalidade, procurando torná-la mais tangível possível para desenvolvedores e designers (Blomkvist, 2002).

Resumidamente, pode-se dizer que dos três critérios adotados para avaliar as heurísticas de acessibilidade, houve um problema isolado de aplicabilidade, algumas sugestões para facilitar o entendimento das heurísticas e, com isto, aumentar a satisfação dos avaliadores e o tempo para aplicar as heurísticas de acessibilidade foi considerado adequado. Portanto, optou-se por mais uma iteração no ciclo de refinamento das heurísticas a fim de se tentar melhorá-las, especialmente na questão da satisfação dos avaliadores.

4.2.2. Segundo experimento

Com base nas sugestões coletadas com os avaliadores no primeiro experimento, uma nova versão das heurísticas foi elaborada (os trechos sublinhados identificam as alterações com relação à versão anterior):

1. **Suporte a diferentes tipos de entrada e saída**

   O sistema deve ser utilizável através dos dispositivos de entrada e saída que o usuário adotar, não importa quais sejam. Ou seja, <u>as funcionalidades presentes no sistema devem ser acessíveis através destes dispositivos</u>. Em especial, o sistema deve suportar a interação por meio de teclado, mouse e leitor de telas.

2. **Conteúdo para todos os usuários**

   Todo conteúdo presente no sistema deve ser compreensível para todos os usuários, ou para o maior número possível de usuários, o que pode incluir muitas vezes fornecê-lo de diferentes formas. Alguns exemplos incluem:

   Figuras: uma descrição textual da mesma (para cegos utilizadores de leitores de tela)

   Músicas e Podcasts: letra ou transcrição (para surdos oralizados)

   Texto escrito: língua de sinais, sinalizada na forma de animação, escrita de sinais ou substituir informação textual por ícones, animações ou outros componentes visuais (para surdos não oralizados)

Vídeos e animações: língua de sinais por meio de um intérprete na tela

Diferenciação de informação por meio de cor, como campos obrigatórios de um formulário: marcação também com uso de símbolos (para pessoas com cromodeficiências).

3. **Independência de uso**

Usuários devem ser capazes de ligar, desligar e configurar de forma autônoma todas as funcionalidades relacionadas à acessibilidade que o sistema fornece. Tais funcionalidades podem ser, por exemplo, aumentar o tamanho da fonte, habilitar o teclado virtual ou a lupa/lente de aumento, alternar para texto com alto contraste de cores, reduzir ou aumentar o tempo de varredura do teclado virtual, entre outros.

4. **Respeito às preferências do usuário**

Usuários possuem diferentes particularidades e habilidades e, desta forma, podem necessitar de configurações específicas para as cores usadas na tela, o volume dos alto-falantes, o tamanho e o tipo de fonte dos textos, o tempo de resposta, entre outros. Portanto, o sistema deve respeitar tais configurações definidas pelo usuário, principalmente se tais configurações foram definidas de forma global (no sistema operacional ou no navegador de Internet, por exemplo).

5. **Eficiência em navegação alternativa**

O sistema deve oferecer maneiras para o usuário evitar percorrer longos caminhos ao usá-lo com tecnologias assistivas. Um sistema pode se mostrar muito ineficiente para os usuários de leitores de telas, teclados virtuais e outras tecnologias assistivas. Por exemplo, ao navegar por um site que não conhecem, muitas vezes os usuários de leitores de telas são obrigados a ler muitos parágrafos antes de chegar ao que efetivamente importa ou desejam. Assim, o sistema deve oferecer teclas de atalho para facilitar o acesso às principais funcionalidades e, no caso de conteúdos textuais, deve procurar agrupá-los de alguma forma, incluir meios de acessar rapidamente estes grupos e informar apenas o necessário por meio de um diálogo simples e direto.

Poucas alterações foram realizadas nas heurísticas, procurando sempre reescrevê-las de acordo com sugestões dos avaliadores, tentando torná-las mais claras. Para tentar contornar a principal dificuldade dos avaliadores durante o experimento anterior, na descrição da primeira heurística, "Suporte a diferentes tipos de entrada e saída", foi incluída a frase "as funcionalidades presentes no sistema devem ser acessíveis através destes dispositivos", ao passo que na descrição da quinta heurística, "Eficiência em navegação alternativa", foi adicionada a frase inicial "O sistema deve oferecer maneiras para o usuário evitar percorrer longos caminhos ao usá-lo com tecnologias assistivas".

Na descrição da segunda heurística, "Conteúdo para todos os usuários", foi incluído o exemplo de conteúdo alternativo para vídeos e animações. E, na descrição da terceira heurística, "Independência de uso", a primeira frase foi reescrita, removendo-se o trecho final "sem a ajuda de terceiros", mas tentando deixar claro que o usuário deveria realizar as configurações de acessibilidade por conta própria, de maneira autônoma, sem auxílio de outras pessoas.

De posse deste conjunto revisado de heurísticas, deu-se então o segundo experimento da fase de refinamento, a ser discutido em detalhes a seguir.

4.2.2.1. Materiais e métodos

O segundo experimento ocorreu de forma similar ao primeiro. Novamente, os mesmos quatro participantes foram recrutados, muito por conta da dificuldade em se encontrar outros potenciais participantes com o perfil desejado (avaliadores de usabilidade) e, também, por se achar apropriado averiguar se o aprendizado prévio foi efetivamente suficiente para uma nova aplicação das heurísticas de acessibilidade. E, seguindo o mesmo processo do primeiro experimento, todos os participantes assinaram um termo de consentimento antes de iniciar as atividades, o mesmo encontrado no Anexo C.

Primeiramente, foi entregue aos participantes uma lista impressa com a nova versão das heurísticas e um documento contendo exemplos de violações das mesmas. Em seguida,

foi realizada a leitura para o grupo de participantes do título e da descrição de cada heurística, explicitando as alterações realizadas nas mesmas e, na medida do possível, relacionando-as com alguns dos problemas de acessibilidade que foram encontrados no experimento anterior e descritos no Anexo D, sempre indagando os participantes sobre potenciais dúvidas. Todo este processo foi realizado em cerca de 30 minutos.

Em seguida, cada um dos participantes sentou-se individualmente em um computador com a tarefa de avaliar a acessibilidade do comunicador instantâneo Windows Live Messenger[20], durante no máximo 45 minutos. Os computadores estavam com sistema operacional Windows XP e o leitor de telas do CPqD – por sinal, apenas um dos participantes já o conhecia. Como no experimento anterior, eu permaneci no laboratório durante todo o tempo a fim de esclarecer dúvidas dos participantes e anotar as dificuldades encontradas na avaliação.

No entanto, antes de iniciarem a avaliação da acessibilidade do sistema proposto, os principais comandos do leitor de telas do CPqD foram repassados aos participantes. Foi dado um tempo de 15 minutos para que os participantes pudessem testar as funcionalidades do leitor de telas antes de efetivamente usá-lo com o Windows Live Messenger. Ainda, uma cópia impressa do manual de uso do leitor de telas do CPqD foi disponibilizada aos participantes, caso desejassem consultá-lo durante a avaliação.

Após a avaliação individual, eu e os participantes novamente nos reunimos para discutir os problemas de acessibilidade encontrados, durante aproximadamente 45 minutos. E, ao final, um questionário muito similar ao do primeiro experimento foi aplicado aos participantes para avaliar a satisfação dos mesmos com as heurísticas propostas. O modelo de questionário pode ser visto no Anexo C.

4.2.2.2. Resultados

Logo durante a leitura das heurísticas de acessibilidade para o grupo de participantes, um deles sugeriu substituir o termo “surdos oralizados” por “surdos

[20] http://download.live.com/messenger. Informação capturada em 11/nov/2009.

alfabetizados" na descrição da heurística "Conteúdo para todos os usuários", por considerar mais adequado. Todavia, o termo "alfabetizado" remete a uma pessoa que passou pelo processo de aprendizado de ler e escrever, mas não necessariamente tal pessoa domina a leitura e a escrita e faz uso frequente e competente das mesmas nas práticas sociais, tornando-se letrada (Soares, 2004). Então, mesmo que seja "alfabetizada", a pessoa pode ter dificuldades para compreender um texto escrito. Por outro lado, o "oralizado" diz respeito à pessoa surda que é capaz de comunicar-se pela língua falada, compreende a língua falada por meio de leitura dos lábios, mas também não necessariamente domina, de fato, a leitura e a escrita. Assim, um termo mais apropriado seria **domínio da língua escrita**, no sentido de tanto ser capaz de ler quanto escrever.

Outro questionamento que veio à tona também no período de leitura das heurísticas foi como seria possível identificar exatamente onde está o problema quando um leitor de telas não lê uma informação relevante: se no programa avaliado, o Windows Live Messenger, ou se no leitor de telas em si. Foi esclarecido, então, que o que se avalia é sempre o conjunto (leitor de telas, sistema operacional e aplicação, no caso, o Windows Live Messenger), da mesma forma que quando se faz uma avaliação heurística de usabilidade também se está avaliando um conjunto formado por software e dispositivos de entrada e saída convencionais, como o monitor, teclado e mouse, é muito difícil separar ou isolar estas variáveis.

No processo de experimentação do leitor de telas do CPqD, os participantes acabaram por escrever algumas frases no Bloco de Notas e no WordPad do Windows XP e não tiveram dificuldades para aprender a usar os comandos básicos de controle da leitura. Três deles anotaram os comandos de pausar, continuar e configurar a leitura em uma folha de papel. O único que não anotou os comandos foi o participante que já conhecia o leitor de telas.

Já na avaliação individual do Windows Live Messenger, todos, sem exceção, tentaram inspecionar o sistema usando, além do leitor de telas, o teclado sem o mouse, as configurações de alto contraste do Windows e o Teclado Virtual do Windows.

Após a avaliação individual, na reunião para discutir os problemas encontrados, os comentários dos participantes foram bem positivos. Todos concordaram que, esta segunda avaliação realizada foi bem mais fácil do que a primeira, sinalizando que a experiência

como avaliadores de acessibilidade também influencia na facilidade ou dificuldade em encontrar os problemas da interface. Inclusive, um dos participantes que mais teve dificuldade no experimento anterior em diferenciar corretamente as heurísticas "Suporte a diferentes tipos de entrada e saída" e "Eficiência em navegação alternativa" comentou que gostou bastante deste segundo experimento, achando que foi mais fácil avaliar e que as heurísticas estavam mais claras do que a primeira vez.

O que se notou, de fato, foi que os participantes não demonstraram quaisquer dificuldades em associar heurísticas de acessibilidade aos problemas encontrados e, ao contrário do que ocorreu durante o primeiro experimento, não sugeriram alterações nas heurísticas e na descrição das mesmas, exceto aquela ocorrida durante a leitura das heurísticas antes de iniciar as avaliações em si.

Ao todo, os participantes encontraram 30 problemas distintos de acessibilidade usando as heurísticas propostas, sendo que a maioria dos problemas foi associada à heurística "Suporte a diferentes tipos de entrada e saída", por conta da não leitura de várias partes do Windows Live Messenger. A Tabela 4.4 identifica a quantidade de problemas encontrados para cada uma das heurísticas e a Tabela 4.5 apresenta a quantidade de problemas encontrados por cada participante. O Anexo E dispõe de uma lista condensada descrevendo todos os problemas encontrados pelos participantes – alguns exemplos de problemas de acessibilidade encontrados podem ser vistos a seguir.

Tabela 4. 4. Problemas encontrados para cada uma das heurísticas no segundo experimento.

Heurística	**Número de ocorrências**
Suporte a diferentes tipos de entrada e saída	18
Conteúdo para todos os usuários	3
Independência de uso	1
Respeito às preferências do usuário	4
Eficiência em navegação alternativa	4

- **Problema:** A única diferença do símbolo indicando quem está online e quem está offline é a cor. **Heurística(s) violada(s):** Conteúdo para todos os usuários. **Participantes que encontraram:** P4.
- **Problema:** Com o leitor de telas, é possível entrar na lista de contatos, mas o mesmo não lê o nome de quem está nela e o status dos contatos (online, offline, ocupado, etc.). **Heurística(s) violada(s):** Suporte a diferentes tipos de entrada e saída. **Participantes que encontraram:** P2, P3 e P4.
- **Problema:** Usuário não consegue aumentar a fonte sozinho caso ele necessite de fontes grandes para conseguir ler as mensagens no Messenger. **Heurística(s) violada(s):** Independência de uso. **Participantes que encontraram:** P3.

Tabela 4. 5. Problemas encontrados por participante no segundo experimento.

Participante	Total de problemas encontrados	Encontrados também por outros participantes
P1	4	2
P2	17	9
P3	10	4
P4	15	9

Usando os dados a respeito dos problemas encontrados da Tabela 4.5, pode-se averiguar que, em média, cada participante descobriu 11,5 problemas de acessibilidade, o que representa aproximadamente um terço do total de problemas encontrados – proporção praticamente idêntica à encontrada no primeiro experimento. Novamente, e analisando a lista completa de problemas encontrados disponível no Anexo E, apenas um dos 30 problemas distintos foi encontrado por todos os participantes. Outros 5 problemas foram encontrados por três dos quatro participantes e 3 problemas por dois dos participantes. Ou seja, e de acordo com os dados condensados na Tabela 4.5, comparando-se com o experimento anterior, houve uma maior sobreposição entre os conjuntos individuais de

problemas encontrados, mas, ainda assim, a maioria dos problemas (21 dos 30 problemas) foi descoberta por apenas um participante, o que continua a reforçar a necessidade da avaliação heurística de acessibilidade ser sempre aplicada por um grupo de avaliadores, da mesma forma como ocorre com a avaliação heurística de usabilidade, pois diferentes avaliadores descobrem diferentes problemas. É muito provável que a maior sobreposição dos conjuntos individuais de problemas encontrados tenha ocorrido pelo fato dos participantes terem um pouco mais de experiência e, com isto, identificaram mais problemas em comum.

Os resultados dos questionários estão disponíveis na Tabela 4.6 e nas Figuras 4.9 a 4.13, novamente considerando a seguinte escala de respostas para as perguntas no questionário:

1. Discordo Totalmente
2. Discordo Parcialmente
3. Indiferente
4. Concordo Parcialmente
5. Concordo Totalmente

Tabela 4. 6. Resultados dos questionários do segundo experimento (média, desvio padrão e moda).

Q	Questão	Média	Desvio Padrão	Moda
1	O tempo para realizar a avaliação com as heurísticas de acessibilidade foi satisfatório.	4,25	0,50	4
2	De forma geral, as heurísticas estavam descritas de forma clara e de fácil compreensão.	5,00	0,00	5
3	Eu entendi do que trata a heurística "Suporte a diferentes tipos de entrada e saída".	5,00	0,00	5
4	Eu entendi do que trata a heurística "Conteúdo para todos os usuários".	4,75	0,50	5
5	Eu entendi do que trata a heurística "Independência de uso".	5,00	0,00	5
6	Eu entendi do que trata a heurística "Respeito às preferências do usuário".	5,00	0,00	5
7	Eu entendi do que trata a heurística "Eficiência em navegação alternativa".	5,00	0,00	5
8	Encontrei problemas graves de acessibilidade usando as heurísticas.	5,00	0,00	5
9	Encontrei problemas de acessibilidade que nem imaginava que existiam usando as heurísticas.	4,75	0,50	5
10	Tentaria usar novamente as heurísticas se fosse necessário avaliar a acessibilidade de outro sistema.	5,00	0,00	5
11	No geral, me senti satisfeito com os resultados da minha avaliação de acessibilidade.	4,75	0,50	5
12	Acredito que posso aprimorar mais minha capacidade de avaliar a acessibilidade usando as heurísticas outras vezes.	5,00	0,00	5

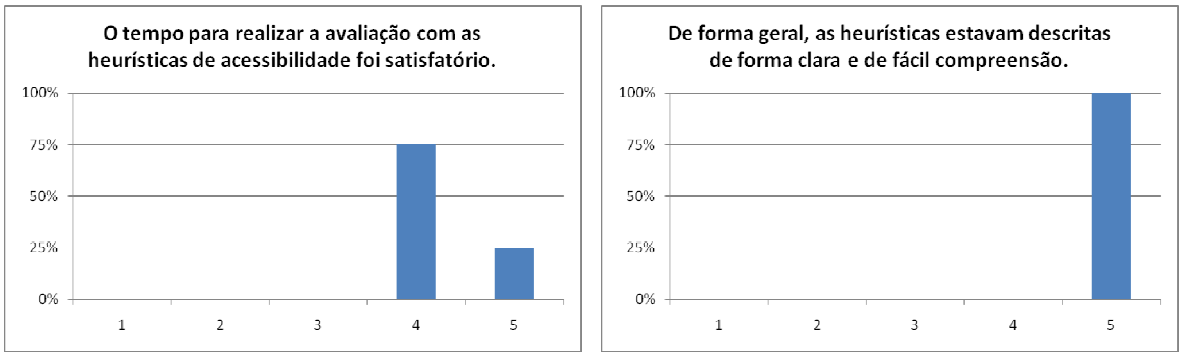


Figura 4. 9. Respostas dos avaliadores para as perguntas sobre o tempo dedicado a uma avaliação e clareza das heurísticas (segundo experimento).

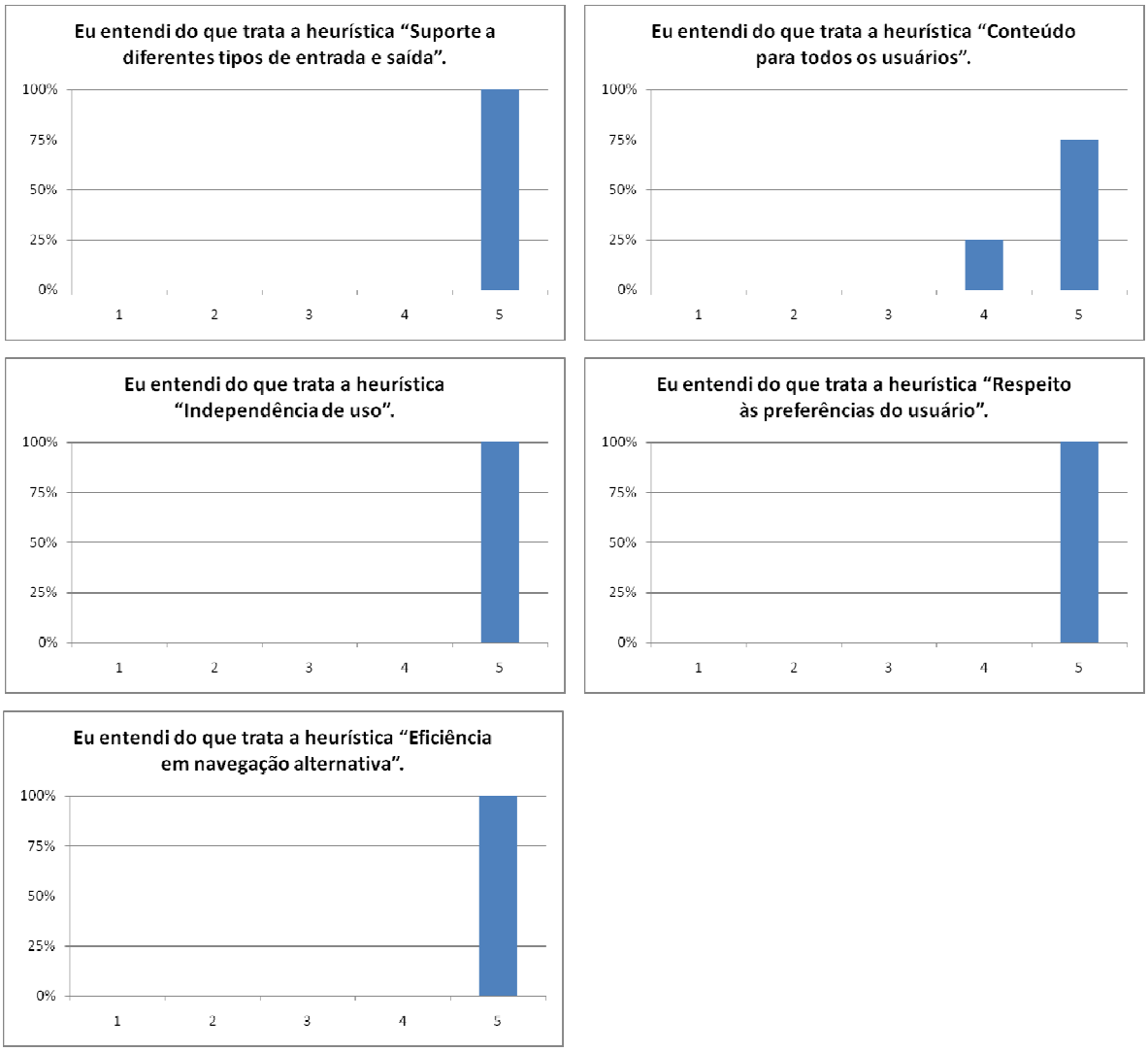


Figura 4. 10. Respostas dos avaliadores para as perguntas sobre o entendimento das heurísticas de acessibilidade (segundo experimento).

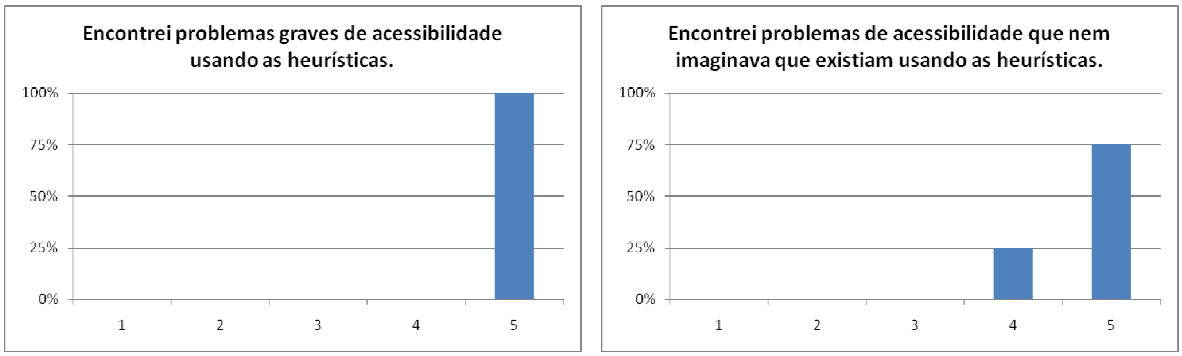


Figura 4. 11. Respostas dos avaliadores para as perguntas sobre os problemas encontrados (segundo experimento).

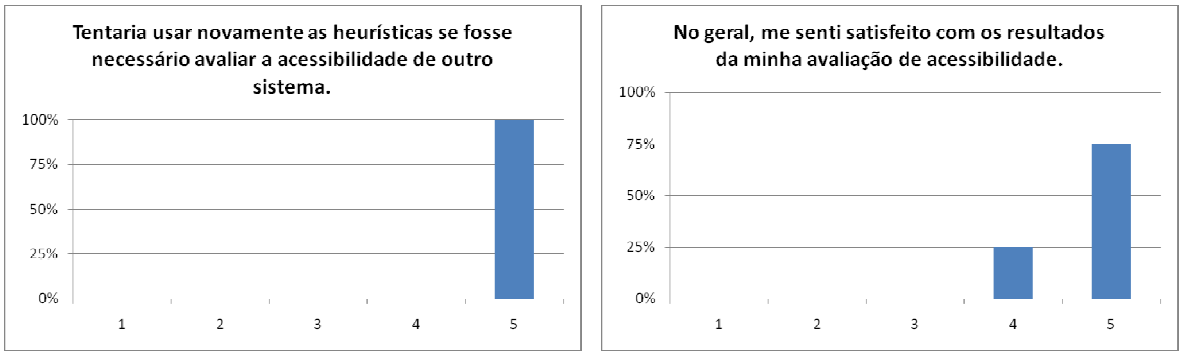


Figura 4. 12. Respostas dos avaliadores para as perguntas sobre a possibilidade de novo uso das heurísticas e a satisfação com os resultados obtidos (segundo experimento).

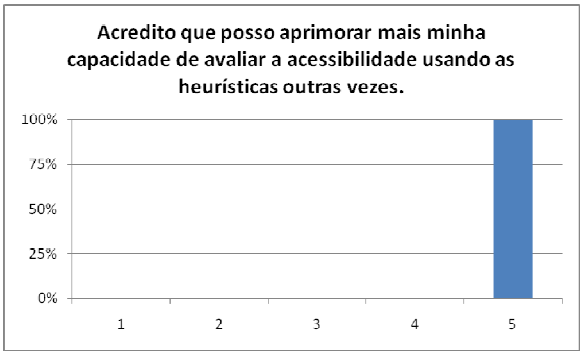


Figura 4. 13. Respostas dos avaliadores para as perguntas sobre a possibilidade de se melhorar as avaliações à medida que se torna mais experiente (segundo experimento).

As respostas dos participantes aos questionários foram totalmente positivas, superior às do primeiro experimento (possivelmente como um resultado do aprendizado prévio das heurísticas no experimento anterior), e muito mais concentradas, o que pode ser observado na Tabela 4.6 pelo valor do desvio padrão reduzido. Novamente, os participantes consideraram adequado o tempo necessário para efetuar uma avaliação de acessibilidade usando as heurísticas propostas, conforme gráfico na Figura 4.9. Adicionalmente, também visto na Figura 4.9, todos julgaram que as heurísticas estavam descritas de forma clara e de fácil compreensão. Diferente do que ocorreu no experimento anterior, todas as heurísticas foram muito bem compreendidas pelos participantes, inclusive a heurística "Eficiência em navegação alternativa", o que pode ser visto nos gráficos da Figura 4.10.

Comparando as respostas ao item "Encontrei problemas de acessibilidade que nem imaginava que existiam usando as heurísticas" nos dois experimentos, observa-se uma diferença significativa nos resultados. No primeiro experimento, dois participantes (50%) deram respostas não satisfatórias, entre 2 (Discordo parcialmente) e 3 (Indiferente), o que já não ocorreu no segundo experimento. Diversas variáveis podem ter contribuído para esta diferença, mas suspeita-se que esteja ligada principalmente às diferenças entre as interfaces avaliadas (Google Docs e Windows Live Messenger). Outras variáveis que poderiam ser consideradas são o nível de experiência prévio dos participantes como usuários de tais interfaces, a reformulação das heurísticas e o nível de compreensão das heurísticas pelos participantes, que certamente foi maior no segundo experimento uma vez que já haviam realizado uma avaliação prévia usando as heurísticas.

Os demais resultados dos questionários se assemelham aos do experimento anterior. Analisando-se os gráficos e as respostas dos questionários, pode-se dizer que os participantes se sentiram satisfeitos com os resultados da avaliação de acessibilidade e acreditam que podem melhorar a capacidade de avaliar a acessibilidade de uma interface usando as heurísticas outras vezes – Figuras 4.12 e 4.13.

Três dos quatro participantes usaram o espaço no questionário destinado à escrita de comentários, críticas e sugestões a respeito da avaliação heurística de acessibilidade. Um deles exaltou como a experiência anterior de uso das heurísticas o ajudou a sentir muito mais facilidade em aplicá-las desta vez. Outro participante comentou sobre o número reduzido de heurísticas, que permite "uma rápida assimilação e uma avaliação eficiente e

eficaz". Já um terceiro participante considerou que nesta versão as heurísticas estavam descritas de forma direta e objetiva, "auxiliando a identificação dos problemas e tornando mais fácil a sua classificação", o que contribuiu também para reduzir confusões na associação dos problemas encontrados com as heurísticas, como ocorridas no experimento anterior. Possivelmente, as opiniões dos participantes a respeito das heurísticas foram influenciadas pelo seu próprio aprendizado a respeito das heurísticas e a respeito de acessibilidade, o que é um fato significativo, uma vez que se espera que, à medida que o avaliador participa de mais avaliações de acessibilidade, o mesmo consiga aprimorar cada vez mais suas avaliações e seu entendimento sobre acessibilidade, da mesma forma como acontece com os avaliadores que aplicam as heurísticas de usabilidade de Nielsen (1994).

Para finalizar, tendo como base os resultados obtidos na avaliação deste segundo experimento, é possível afirmar que, diferente do primeiro experimento, não ocorreram problemas relacionados ao critério aplicabilidade e os avaliadores se sentiram mais satisfeitos com o método de avaliação em si, sendo que o tempo necessário para avaliação também foi considerado adequado. Portanto, e aparentemente, as heurísticas estão aptas para o próximo passo, que é exatamente sua aplicação em avaliações reais de acessibilidade. E, assim como ocorreram com as heurísticas de usabilidade de Nielsen (1994), poderão passar por uma revisão, refletindo sugestões e críticas de outros pesquisadores, especialistas, designers e desenvolvedores que as adotarem.

4.3. Versão final das heurísticas de acessibilidade

Com os resultados positivos do segundo ciclo da fase de refinamento, chega-se à versão final das heurísticas de acessibilidade – as alterações no texto com relação à versão anterior estão sublinhadas:

1. **Suporte a diferentes tipos de entrada e saída**

   O sistema deve ser utilizável através dos dispositivos de entrada e saída que o usuário adotar, não importa quais sejam. Ou seja, as funcionalidades presentes no sistema

devem ser acessíveis através destes dispositivos. Em especial, o sistema deve suportar a interação por meio de teclado, mouse e leitor de telas.

2. **Conteúdo para todos os usuários**

   Todo conteúdo presente no sistema deve ser identificável por todos os usuários, ou para o maior número possível de usuários, o que pode incluir muitas vezes fornecê-lo de diferentes formas. Alguns exemplos incluem:

   Figuras: uma descrição textual da mesma (para cegos utilizadores de leitores de tela)

   Músicas e Podcasts: letra ou transcrição (para surdos que dominam a língua escrita)

   Texto escrito: língua de sinais, sinalizada na forma de animação, escrita de sinais ou substituir informação textual por ícones, animações ou outros componentes visuais (para surdos que não dominam a língua escrita)

   Vídeos e animações: língua de sinais por meio de um intérprete na tela

   Diferenciação de informação por meio de cor, como campos obrigatórios de um formulário: marcação também com uso de símbolos (para pessoas com cromodeficiências).

3. **Independência de uso**

   Usuários devem ser capazes de ligar, desligar e configurar de forma autônoma todas as funcionalidades relacionadas à acessibilidade que o sistema fornece. Tais funcionalidades podem ser, por exemplo, aumentar o tamanho da fonte, habilitar o teclado virtual ou a lupa/lente de aumento, alternar para texto com alto contraste de cores, reduzir ou aumentar o tempo de varredura do teclado virtual, entre outros.

4. **Respeito às preferências do usuário**

   Usuários possuem diferentes particularidades e habilidades e, desta forma, podem necessitar de configurações específicas para as cores usadas na tela, o volume dos alto-falantes, o tamanho e o tipo de fonte dos textos, o tempo de resposta, entre outros. Portanto, o sistema deve respeitar tais configurações definidas pelo usuário, principalmente se tais configurações foram definidas de forma global (no sistema operacional ou no navegador de Internet, por exemplo).

5. **Eficiência em interação alternativa**

O sistema deve oferecer maneiras para o usuário evitar percorrer longos caminhos ao usá-lo com tecnologias assistivas. Um sistema pode se mostrar muito ineficiente para os usuários de leitores de telas, teclados virtuais e outras tecnologias assistivas. Por exemplo, ao navegar por um site que não conhecem, muitas vezes os usuários de leitores de telas são obrigados a ler muitos parágrafos antes de chegar ao que efetivamente importa ou desejam. Assim, o sistema deve oferecer teclas de atalho para facilitar o acesso às principais funcionalidades e, no caso de conteúdos textuais, deve procurar agrupá-los de alguma forma, incluir meios de acessar rapidamente estes grupos e informar apenas o necessário por meio de um diálogo simples e direto.

Três alterações realizadas foram na segunda heurística, "Conteúdo para todos os usuários", na qual se substituiu os termos "surdos (não) oralizados" por "surdos que (não) dominam a língua escrita", após análise da sugestão realizada por um dos participantes e também a consulta a uma especialista em educação de surdos, e o termo "compreensível" foi substituído por "identificável", visto que há uma diferença significativa entre conseguir identificar e conseguir compreender – há inúmeros fatores relacionados à compreensão (idioma, nível de escolaridade do usuário, especificidade e contexto do sistema, experiência do usuário, entre outros), que tornam praticamente intangível atender esta heurística. Também, na quinta heurística, a palavra "navegação" do título foi substituída por "interação" para dar uma idéia de um contexto mais amplo de aplicação da heurística e não apenas navegação pela Web como alguém poderia imaginar. O restante do texto das heurísticas foi mantido inalterado.

Para oferecer uma consulta rápida, a mesma listagem da versão final das heurísticas está disponível nas últimas páginas desta tese de doutorado, no Anexo F.

Capítulo 5

Conclusões

Avaliar a acessibilidade tem sido uma atividade onerosa em qualquer processo de desenvolvimento de software. Métodos tradicionalmente adotados, os testes com usuários com deficiências e a verificação de conformidade com *guidelines*, auxiliada ou não por ferramentas semi-automáticas, demandam uma quantidade considerável de recursos (tempo, recrutamento de pessoas, treinamento nos métodos de avaliação, etc.), muito superior se comparados com os métodos de avaliação de usabilidade.

Não bastasse a questão dos custos, outros aspectos impactam negativamente nos dois métodos tradicionais de avaliação de acessibilidade, conforme visto no Capítulo 3. As principais *guidelines* existentes hoje em dia, as WCAG 1.0 são difíceis de serem aplicadas fora do contexto da Web e sofrem críticas ferrenhas quanto à sua complexidade, completude e a validade de determinadas *guidelines*. E mesmo quando ferramentas semi-automáticas são usadas para auxiliar a verificação das *guidelines*, o tempo necessário para realizar a avaliação é um tanto quanto elevado (média de uma hora por página Web). Já os testes com usuários com deficiência demandam uma quantidade de participantes diretamente proporcional ao nível de universalidade que se almeja alcançar, além de inevitáveis adaptações nos materiais e métodos consagrados para testes com usuários (pensar em voz alta, questionários, entre outros).

Há, de fato, outros métodos alternativos propostos na literatura para avaliar a acessibilidade, como os percursos de Brajnik (2006) e de Kato e Hori (2006) e as heurísticas de Paddison e Englefield (2003) e de Koivunen e McCathieNevile (2001). Todavia, nem todos os autores de novos métodos de avaliação de acessibilidade realizaram alguma avaliação de seus métodos, considerando critérios como aplicabilidade, tempo para aplicação (e conseqüentemente o custo) e a satisfação dos avaliadores – de fato, alguns sequer realizaram qualquer avaliação do método que propuseram. Adicionalmente, existem

casos como os de Paddison e Englefield e de Koivunen e McCathieNevile nos quais o método de avaliação proposto continua restrito a contextos da Web por terem sido baseados nas WCAG 1.0.

Não restam dúvidas de que há uma carência por um método de avaliação de acessibilidade barato, fácil de aprender e rápido de aplicar e que possa avaliar efetivamente qualquer sistema, seja ele para a Web ou não. Um método de avaliação de acessibilidade com tais características, que são praticamente as mesmas características apresentadas pela avaliação heurística de Nielsen (1994), por sinal, o método de inspeção de usabilidade mais consagrado e adotado atualmente, seria muito menos intimidador para qualquer desenvolvedor e sofreria muito menos restrições para ser adotado por qualquer processo de desenvolvimento de software.

As heurísticas de acessibilidade apresentadas nesta tese de doutorado foram concebidas exatamente para tentar preencher esta lacuna. Primeiro, o conjunto de heurísticas de acessibilidade proposto foi resultado de um processo que abstraiu o contexto de execução do sistema, sendo, portanto, aplicável a diferentes tipos de sistemas, mesmo àqueles que não são baseados na Web. Segundo, as heurísticas de acessibilidade propostas passaram por uma avaliação visando atender a critérios como o tempo de aplicação e a facilidade de compreensão das mesmas, satisfazendo os interesses de avaliadores e desenvolvedores de software e facilitando a apropriação do método de avaliação de acessibilidade pelos mesmos. Terceiro, uma vez que o conjunto de heurísticas é facilmente aprendido e de rápida aplicação, possibilita avaliar a acessibilidade durante todo o ciclo de desenvolvimento de um software, como é recomendado pelas boas práticas da área de IHC. Quarto, e diferente de outras heurísticas para avaliação de interface, seja de usabilidade, seja de acessibilidade, todo o processo de geração das heurísticas foi documentado por meio desta tese de doutorado.

É esperado também que, assim como ocorre com as heurísticas de Nielsen para usabilidade, as heurísticas de acessibilidade propostas contribuam para melhorar o entendimento a respeito de acessibilidade dos próprios avaliadores à medida que realizam avaliações e compartilham suas descobertas com outros avaliadores.

Embora o método de avaliação proposto possua diversas qualidades, há um ponto que deve ser ressaltado: a avaliação heurística de acessibilidade não pode e nem deve ser

encarada como o método definitivo para avaliação de acessibilidade. Da mesma forma como ocorre com os métodos de avaliação de usabilidade, é recomendável sempre a aplicação de outros métodos, especialmente os empíricos, envolvendo a participação de potenciais usuários com deficiências, de forma a complementar os resultados e se obter uma boa cobertura de problemas encontrados. Também, em alguns casos, especialmente quando se trata de sites governamentais, pode haver alguma obrigação legal de se seguir um conjunto de *guidelines* específicas, o que inevitavelmente leva a uma avaliação de conformidade com *guidelines* e que não pode ser substituída por outros métodos de avaliação.

Mesmo com resultados significativos encontrados nesta tese de doutorado, há ainda algumas trilhas que podem e devem ser seguidas para dar prosseguimento à pesquisa. Uma delas é exatamente uma das próximas atividades já planejadas: aplicar as heurísticas de acessibilidade para avaliar alguns ambientes de aprendizado eletrônico, como o TelEduc[21], o Moodle[22] e o TIDIA-Ae[23], comparando os resultados com os de outras avaliações já realizadas – inclusive, o autor desta tese de doutorado já efetuou a verificação de conformidade com as WCAG 1.0 dos três ambientes em questão.

De forma semelhante e não menos importante, com as heurísticas propostas finalmente será possível aplicar efetivamente um método de inspeção de acessibilidade para avaliar o HagáQuê, uma das grandes motivações desta tese de doutorado. E como existem dados a respeito de testes com crianças com deficiências, também será possível realizar comparações entre os métodos.

Outro trabalho futuro muito relevante, que pode ser iniciado a partir dos resultados obtidos das avaliações dos ambientes de aprendizado eletrônico e do HagáQuê, é o estudo sobre a completude de métodos de avaliação de acessibilidade: testes com usuários, verificação de conformidade com *guidelines* e a avaliação heurística de acessibilidade. Um estudo deste tipo seria similar ao de Law e Hvannberg (2002) para métodos de avaliação de usabilidade. Ainda, poderia servir de base para se levantar que problemas tipicamente são

[21] http://www.teleduc.org.br. Informação capturada em 13/nov/2009.
[22] http://moodle.org. Informação capturada em 13/nov/2009.
[23] http://tidia-ae.incubadora.fapesp.br/portal. Informação capturada em 13/nov/2009.

encontrados por cada um dos métodos de avaliação de acessibilidade e a severidade dos mesmos.

Adicionalmente, e apesar já ter me enveredado em alguns estudos que realizaram comparações entre métodos de avaliação de usabilidade e de acessibilidade (Tanaka et al. 2005; Tanaka et al., 2006), ainda existe um grande campo para discutir a relação entre acessibilidade e usabilidade. E mesmo para discutir grandes questões, como se um realmente design universal, para todos, é factível.

Por fim, espera-se que o resultado desta tese de doutorado seja uma contribuição efetiva e real para o design de um mundo mais inclusivo e menos exclusivo.

Referências

Acessibilidade Brasil (2009) DaSilva – Avaliador de Acessibilidade para Websites. Disponível em http://www.dasilva.org.br. Informação capturada em 17/ago/2009.

Adobe. (2009) Adobe Flash accessibility design guidelines. Disponível em: http://www.adobe.com/accessibility/products/flash/best_practices.html. Informação capturada em 11/set/2009.

Alexander, V.; Benson, C.; Cameron, B.; Haneman, B.; O'Briain, P.; Snider, S. (2008) GNOME Accessibility Developers Guide. Disponível em: http://library.gnome.org/devel/accessibility-devel-guide/nightly/. Informação capturada em 11/set/2009.

ATRC (2009) AChecker – Web Accessibility Checker. Disponível em http://achecker.ca. Informação capturada em 11/ago/2009.

Bailey, J.; Burd, E. (2006) What is the current state of Web Accessibility? In Proceedings of the Eighth IEEE international Symposium on Web Site Evolution (September 23 – 24, 2006). WSE. IEEE Computer Society, Washington, DC, 69-74.

Belder, B. (2002) eEurope 2002: Accessibility of Public Web Sites and their Content. Disponível em http://www.europarl.europa.eu/sides/getDoc.do?pubRef=-//EP//TEXT+REPORT+A5-2002-0147+0+DOC+XML+V0//EN. Informação capturada em 20/ago/2009.

Bergman, E.; Johnson, E. (1995) Towards Accessible Human-Computer Interaction, In Advances in Human-Computer Interaction, vol. 5, Nielsen, J. (ed.), Ablex Publishing Corporation, 1995.

Bim, S. A.; Tanaka, E. H.. Rocha, H. V. (2000) HagáQuê – Editor de Histórias em Quadrinhos. In Anais do VI Workshop de Informática na Escola (VI WIE) do XX Congresso da Sociedade Brasileira de Computação (XX SBC 2000), Curitiba/PR, 2000.

Bim, S. A. (2001) HagáQuê – Editor de história em quadrinhos. Dissertação de mestrado. Instituto de Computação, UNICAMP, Campinas/SP, 2001.

Blomkvist, S. (2002) Personas – an overview. Extract from the paper "The User as a Personality. Using Personas as a tool for design". Uppsala University, Dept of IT-HCI, September, 2002. Disponível em http://www.it.uu.se/edu/course/homepage/hcinet/ht04/library/docs/Persona-overview.pdf. Informação capturada em 19/nov/2009.

Borges, J. A. (2002ª) Projeto DOSVOX. Disponível em http://intervox.nce.ufrj.br/dosvox/. Informação capturada em 23/out/2009.

Borges, J. A. (2002b) Projeto Motrix. Disponível em http://intervox.ufrj.br/motrix/. Informação capturada em 23/out/2009.

Brajnik, G. (2006) Web accessibility testing: when the method is the culprit. In Proceedings of the 10th International Conference on Computers Helping People with Special Needs (ICCHP 2006), Springer-Verlag, 2006.

Brajnik, G. (2009) Barrier Walkthrough – Heuristic evaluation guided by accessibility barriers. Disponível em: http://sole.dimi.uniud.it/~giorgio.brajnik/projects/bw/bw.html. Informação capturada em 24/set/2009.

Caldwell, B.; Cooper, M.; Reid, L. G.; Vanderheiden, G. (2008) Web Content Accessibility Guidelines (WCAG) 2.0. Disponível em http://www.w3.org/TR/WCAG20/. Informação capturada em 17/jul/2009.

Chisholm, W.; Vanderheiden, G.; Jacobs, I (1999) Web Content Accessibility Guidelines 1.0. Disponível em: http://www.w3.org/TR/WCAG10/. Informação capturada em 17/jul/2009.

Chisnell, D. (2009) Where do heuristics come from? Disponível em: http://usabilitytestinghowto.blogspot.com/2009/05/where-do-heuristics-come-from.html. Informação capturada em 17/jul/2009.

Choi, Y. S.; Yi, J. S.; Law, C. M.; Jacko, J. A. (2006) Are "Universal Design Resources" Designed for Designers? In Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Accessibility (Portland, Oregon, USA, October 23 – 25, 2006). Assets '06. ACM, New York, NY, 87-94.

Clark, J. (2006) To Hell with WCAG 2. Disponível em: http://www.alistapart.com/articles/tohellwithwcag2. Informação capturada em 20/jul/2009.

Clark, J. (2007) Letter to Tim Berners-Lee: Time to cancel WCAG 2. Disponível em: http://joeclark.org/access/webaccess/WCAG/TBL/. Informação capturada em 20/jul/2009.

CTIC (2009) TAW3 Online. Disponível em http://www.tawdis.net/ingles.html?lang=en. Informação capturada em 11/ago/2009.

Cooper, M.; Rejmer, P. (2001) Case study: localization of an accessibility evaluation. In Proceedings of SIGCHI Conference on Human Aspects in Computing Systems (CHI'01) (Seattle, WA, United States, March 31 – April 5, 2001), ACM Press, New York, NY, 2001, 141-142.

Connell, B. R.; Jones, M.; Mace, R.; Mueller, J.; Mullick, A.; Ostroff, E.; Sanford, J.; Steinfeld, E.; Story, M.; Vanderheiden, G. (1997) The principles of Universal Design. Disponível em: http://design.ncsu.edu/cud/about_ud/udprinciplestext.htm. Informação capturada em 20/ago/2009.

Condigitais (2009) Projeto básico de chamada pública para apoio financeiro à produção de conteúdos educacionais digitais multimídia. Disponível em: http://www.ggpe.prpg.unicamp.br/multimidia/multindex1.htm. Informação capturada em 15/out/2009.

Convington, G. A.; Hannah, B. (1997) Access by Design. John Wiley & Sons, 1997.

Cuperschmid, A. R. M. (2008) Heurísticas de jogabilidade para jogos de computador. Dissertação de Mestrado. Instituto de Artes/UNICAMP, Campinas/SP, 2008.

Dictionary.com (2009ª) Heuristic Definition. Disponível em: http://dictionary.reference.com/browse/heuristic. Informação capturada em 20/jul/2009.

Dictionary.com (2009ª) Guideline Definition. Disponível em: http://dictionary.reference.com/browse/guideline. Informação capturada em 20/jul/2009.

DRA (2006) National Federation of the Blind v. Target. Disponível em: http://www.dralegal.org/cases/private_business/nfb_v_target.php. Informação capturada em 20/ago/2009.

Eduplace (2009a) GeoNet. Disponível em: http://www.eduplace.com/geonet/splash.html. Informação capturada em 12/nov /2009.

Eduplace (2009b) Pre-Season Puzzles. Disponível em: http://www.eduplace.com/tacklereading/puzzles.html. Informação capturada em 12/nov/2009.

Erigami (2009) Truwex Online 2.0 beta. Disponível em: http://checkwebsite.erigami.com/accessibility.html. Informação capturada em 17/ago/2009.

Free Dictionary (2009a) heuristic – definition of heuristic by the Free Dictionary Online. Disponível em: http://www.thefreedictionary.com/heuristic. Informação capturada em 20/jul/2009.

Free Dictionary (2009b) guideline – definition of guideline by the Free Dictionary Online. Disponível em: http://www.thefreedictionary.com/guideline. Informação capturada em 20/jul/2009.

Fujitsu (2009) Fujitsu Web Accessibility Inspector 5.01. Disponível em http://www.fujitsu.com/global/accessibility/assistance/wi/. Informação capturada em 17/ago/2009.

Goldsmith, S. (2001) Designing for the Disabled: The New Paradigm. Architectural Press, Great Britain, 2001.

Governo Eletrônico (2005) eMAG, Acessibilidade de Governo Eletrônico – Cartilha Técnica. Disponível em http://www.governoeletronico.gov.br/acoes-e-projetos/e-MAG. Informação capturada em 27/ago/2009.

Gueury, M. (2009) HTML Validator for Firefox and Mozilla. Disponível em http://users.skynet.be/mgueury/mozilla/. Informação capturada em 17/ago/2009.

HagáQuê (2001) HagáQuê – Editor de história em quadrinhos. Disponível em http://www.nied.unicamp.br/~hagaque. Informação capturada em 20/out/2009.

Henry, S. L. (2005) Introduction to Web Accessibility. Disponível em http://www.w3.org/WAI/intro/accessibility.php. Informação capturada em 20/ago/2009.

Holzinger, A. (2005) Usability Engineering Methods for Software Developers, In Communications of the ACM, Vol. 48, n. 1, Janeiro/2005.

HREOC (2009) World Wide Web Access: Disability Discrimination Act Advisory Notes – Version 3.3.1. Disponível em http://www.hreoc.gov.au/disability_rights/standards/www_3/www_3.html. Informação capturada em 27/ago/2009.

Hull, L. (2004) Accessibility: it's not just for disabilities any more. In interactions, vol. 11, n. 2, ACM Press, New York, NY, Mar, 2004, 36-41.

IBGE (2003) Censo Demográfico 2000 – Tabulação Avançada. População residente, por situação do domicílio e sexo, segundo o tipo de deficiência, Brasília/DF, 2003. Disponível em: http://www.ibge.gov.br/home/estatistica/populacao/censo2000/populacao/tabela_brasil.shtm. Informação capturada em 20/ago/2009.

IBM (2008) Human Ability and Accessibility Center – Developer guidelines. Disponível em: http://www-03.ibm.com/able/guidelines/index.html. Informação capturada em 11/set/2009.

IDT (2009) Tipos de deficiência. Disponível em: http://www.idt.org.br/idt/deficientes/texto.asp?id=23. Informação capturada em 29/dez/2009.

IES (2009) Projeto Instituição de Ensino / Sociedade / Portadores de Deficiência. Disponível em: http://ies.portadoresdedeficiencia.vilabol.uol.com.br/. Informação capturada em 29/dez/2009.

ISO (1998) Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability.

Jacobs, I.; Gunderson, J.; Hansen, E. (2002) User Agent Accessibility Guidelines 1.0. Disponível em: http://www.w3.org/TR/WAI-USERAGENT/. Informação capturada em 17/jul/2009.

Juliato, M.; Tanaka, E. H.; Baudet, C.; Galves, M.; Coelho, T. T.; Rocha, H. V. (2004ª) Tflex: Um Simulador de Teclado com Múltiplos Modos de Varredura. In Proceedings of the VI Symposium on Human Factors in Computing Systems (IHC2004), Curitiba/PR.

Juliato, M.; Tanaka, E. H.; Baudet, C.; Galves, M.; Coelho, T. T.; Rocha, H. V. (2004b) Tflex: Proposta de Simulador de Teclado com Diferentes Varreduras e Layouts, In

Proceedings of the VII Ibero-American Congress on Computers in Education (RIBIE2004), Monterrey, Mexico.

Kato, T.; Hori, M. (2006) "Beyond Perceivability": Critical Requirements for Universal Design of Information. In Proceedings of the 8th international ACM SIGACCESS Conference on Computers and Accessibility (Portland, Oregon, USA, October 23 – 25, 2006). Assets '06. ACM, New York, NY, 287-288.

Kelly, B., Sloan, D., Phipps, L., Petrie, H. and Hamilton, F. (2005) Forcing Standardization or Accommodating Diversity? A Framework for Applying the WCAG in the Real World. In Proceedings of the 2005 international Cross-Disciplinary Workshop on Web Accessibility (W4A 2005) (Chiba, Japan, May 10 – 10, 2005). W4A '05, vol. 88. ACM, New York, NY, 46-54.

Kelly, B.; Sloan, D.; Brown, S.; Seale, J.; Petrie, H.; Lauke, P.; Ball, S. (2007) Accessibility 2.0: People, Policies and Processes. In Proceedings of the 2007 International Cross-Disciplinary Conference on Web Accessibility (W4A 2007) (Banff, Canada, May 07 – 08, 2007). W4A '07, vol. 225. ACM, New York, NY, 138-147.

Kennedy, S. (1989) Using Video in the BNR Usability Lab. In SIGCHI Bull. 21, 2 (Oct. 1989), 92-95.

Koivunen, M. R.; McCathieNevile, C. (2001) Accessible Graphics and Multimedia on the Web. Disponível em: http://www.w3.org/2001/05/hfweb/heuristics.htm. Informação capturada em 20/jul/2009.

Law, L. C.; Hvannberg, E. T. (2002) Complementarity and Convergence of Heuristic Evaluation and Usability Test: A Case Study of UNIVERSAL Brokerage Platform. In Proceedings of 2nd Nordic Conference on Human-Computer Interaction (NordiCHI'02), Aarhus, Dinamarca, Outubro/2002, ACM, New York, NY, 71-80.

Lawson, B. (2006) WCAG 2.0: when I want a beer, don't give me shandy. Disponível em: http://www.brucelawson.co.uk/2006/wcag-20-beer-shandy/. Informação capturada em 11/ago/2009.

Lazar, J. (2001) User-Centered Web Development. Jones & Bartlett Publishers, 2001.

Lewis, C.; Rieman, J. (1994) Task-Centered User Interface Design. Disponível em http://hcibib.org/tcuid/. Informação capturada em 11/out/2009.

Lewis, C. (2006) HCI and Cognitive Disabilities. In Interactions, vol. 13, 3 (May. 2006), ACM Press, New York, NY, 14-15.

Lepistö, A.; Ovaska, S. (2004) Usability evaluation involving participants with cognitive disabilities. In Proceedings of the Third Nordic Conference on Human-Computer interaction (Tampere, Finland, October 23 – 27, 2004). NordiCHI '04, vol. 82. ACM, New York, NY, 305-308.

Leporini, B.; Paternò, F. (2004) Increasing usability when interacting through screen readers. In Universal Access in the Information Society, vol. 3, number 1, Springer Berlin, Heidelberg, March, 2004, 57-70.

Mace, R. (1998) A Perspective on Universal Design. Designing for the 21st Century: An International Conference on Universal Design. Disponível em: http://www.design.ncsu.edu/cud/about_us/usronmacespeech.htm. Arquivo capturado em 20/ago/2009.

MacKay, D.; Vertanen, K. (2005) Dasher – writing fast and free with any muscle. Disponível em http://www.inference.phy.cam.ac.uk/dasher/Publications.html. Informação capturada em 23/out/2009.

MEC (2009) Portal do Professor. Disponível em: http://portaldoprofessor.mec.gov.br. Informação capturada em 23/out/2009.

Miller, G. A. (1956) The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. In The Psychological Review, vol. 63, pp. 81-97. Disponível em: http://www.musanim.com/miller1956/. Informação capturada em 17/jul/2009.

Muller, M. J.; McClard, A.; Bell, B.; Dooley, S.; Meiskey, L.; Meskill, J. A.; Sparks, R.; Tellam, D. (1995) Validating an Extension to Participatory Heuristic Evaluation: Quality of Work and Quality of Work Life. In Proceedings of Conference on Human Factors in Computing Systems (CHI'95), Denver, CO, Estados Unidos, Maio/1995.

Nielsen, J. (1993) Usability Engineering, Academic Press, 1993.

Nielsen, J. (1994) "Heuristic Evaluation", In Usability Inspection Methods, Cap. 2, Nielsen, J. e Mack, R. (ed.), John Wiley & Sons, p. 25-62, 1994.

Nielsen, J. (2000) Why You Only Need to Test with 5 Users. Disponível em: http://www.useit.com/alertbox/20000319.html. Informação capturada em 11/out/2009.

Obrenovic, Z.; Abascal, J.; Starcevic, D. (2007) Universal Accessibility as a Multimodal Design Issue. In Communications of the ACM, vol. 50, n. 5, ACM Press, New York, NY, May, 2007, 83-88.

OMS (2009) International Classification of Functioning, Disability and Health. Disponível em: http://www.who.int/classifications/icf/en/index.html. Informação capturada em 29/dez/2009.

Paddison, C.; Englefield, P. (2003) Applying heuristics to perform a rigorous accessibility inspection in a commercial context. In Proceedings of the 2003 Conference on Universal Usability (Vancouver, British Columbia, Canada, November 10 – 11, 2003). CUU '03. ACM, New York, NY, 126-133.

Pidgin (2009) Pidgin, the universal chat client. Disponível em: http://www.pidgin.im/. Informação capturada em 12/nov/2009.

Pinelle, D.; Wong, N.; Stach, T. (2008) Heuristic evaluation for games: usability principles for video game design. In Proceeding of the Twenty-Sixth Annual SIGCHI Conference on Human Factors in Computing Systems (Florence, Italy, April 05 – 10, 2008). CHI '08. ACM, New York, NY, 1453-1462.

Pólya, G. (1975) How to solve it – a new aspect of mathematical method, 2a. ed., Princeton University Press.

Popov, B. (2006) Can WCAG 2.0 be simpler? Disponível em: http://www.evolt.org/can_wcag_2_be_simpler. Informação capturada em 20/jul/2009.

Powlik, J. J.; Karshmer, A. J. (2002) When accessibility meets usability. In Universal Access in the Information Society, vol. 1, number 3, june, 2002, Springer Berlin/Heidelberg, 217-222.

Prates, R.O.; Barbosa, S.D.J. (2003) Avaliação de Interfaces de Usuário – Conceitos e Métodos. In Anais do XXIII Congresso Nacional da Sociedade Brasileira de Computação. XXII Jornadas de Atualização em Informática (JAI). SBC'2003. Agosto de 2003.

Pressman, R. (2001) Software Engineering – a Practitioner's Approach. 5th 146e, McGraw Hill, 2001.

Proinesp (2001) Proinesp – Projeto de Informática na Educação Especial. Disponível em: http://www.nied.unicamp.br/~proinesp/. Informação capturada em 20/out /2009.

Puchkin, V. N. (1976) Heurística: a ciência do pensamento criador, 2ª. 147e., Zahar Editores.

Usability.gov (2006) Research-based Web Design & Usability Guidelines. Disponível em: http://www.usability.gov/pdfs/guidelines.html. Informação capturada em 20/jul/2009.

RIVED (2009) Rede Interativa Virtual de Educação. Disponível em: http://rived.mec.gov.br/site_objeto_lis.php. Informação capturada em 23/out/2009.

Rocha, H. V.; Baranauskas, M. C. C. (2003) Design e Avaliação de Interfaces Humano-Computador, Núcleo de Informática Aplicada à Educação, 2003.

Rosson, M.B.; Carroll, J.M. (2002) Usability Engineering: Scenario-Based Development of Human-Computer Interaction. Morgan Kaufman, New York, NY, 2002.

Rybená (2009) Rybená Player. Disponível em: http://www.rybena.org.br. Informação capturada em 11/nov/2009.

Santana, A. L. (2007) Deficiência Mental. Disponível em: http://www.infoescola.com/psicologia/deficiencia-mental/. Informação capturada em 29/dez/2009.

Section 508 (2000) Section 508 Standards. Disponível em: http://www.section508.gov/. Informação capturada em 20/jul/2009.

Soares, M. B. (2004) O que é letramento e alfabetização. Disponível em: http://www.moderna.com.br/moderna/didaticos/ef1/artigos/2004/0014.htm. Informação capturada em 19/nov/2009.

Somervell, J.; Wahid, S.; McCrickard, D. S. (2003) Usability Heuristics for Large Screen Information Exhibits. In Proceedings of the INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction (Zurich, Switzerland, September 1st-5th,2003). IOS Press 2003, 904-907.

Somervell, J.; McCrickard, S. (2005) Better discount evaluation: illustrating how critical parameters support heuristic creation. In Interacting with Computers, Volume 17, Issue 5, Social Impact of Emerging Technologies, July 2005, Pages 592-612.

Schwerdtfeger, R. S. (2006) IBM Guidelines for Writing Accessible Applications Using 100% Pure Java – version 2.4. Disponível em http://www-03.ibm.com/able/guidelines/java/snsjavag.html. Informação capturada em 11/set/2009.

Tanaka, E. H. (2004) Tornando um software acessível às pessoas com necessidades educacionais especiais. Dissertação de mestrado, Instituto de Computação, UNICAMP, Campinas/SP, 2004.

Tanaka, E. H.; Bim, S. A.; Rocha, H. V. (2005) Comparing accessibility evaluation and usability evaluation in HagáQuê. In Proceedings of the 2005 Latin American Conference on Human-Computer interaction (Cuernavaca, Mexico, October 23 – 26, 2005). CLIHC '05, vol. 124. ACM, New York, NY, 139-147.

Tanaka, E. H.; Vargas, A.; Silva, A. C.; Rocha, H. V. (2006) Centenas de razões para achar o JEMS difícil. In Anais do VII Simpósio sobre Fatores Humanos em Sistemas Computacionais (IHC 2006), Competição de avaliação de sistemas, Natal/RN.

Termens, M.; Ribera, M.; Porras, M.; Boldú, M.; Sulé, A.; Paris, P. (2009) Web Content Accessibility Guidelines: from 1.0 to 2.0. In Proceedings of the 18th international Conference on World Wide Web (Madrid, Spain, April 20 – 24, 2009). WWW '09. ACM, New York, NY, 1171-1172.

Theofanos, M. F.; Redish, J. (2003) Bridging the gap: between accessibility and usability. In Interactions, vol. 10, n. 6, ACM Press, New York, NY, November+December, 2003, 36-51.

Treviranus, J; McCathieNevile, C; Jacobs, I; Richards, J. (2000) Authoring Tool Accessibility Guidelines 1.0. Disponível em: http://www.w3.org/TR/ATAG10/. Informação capturada em 17/jul/2009.

Usability Junction. (2002) What is accessibility? Johannesburgo, África do Sul, 2002. Disponível na em: http://www.usabilityjunction.com/services/faq_access.htm. Arquivo capturado em 30/dez/2002.

Vizim, M. (2003) Educação Inclusiva: O Avesso e o Direito de uma mesma realidade. In: Políticas Públicas: Educação, Tecnologias e Pessoas com Deficiências, Silva, S.; Vizim, M. (orgs.), Mercado de Letras: Associação de Leitura do Brasil, Campinas/SP, p. 49-71, 2003.

WAI (2009a) Complete List of Web Accessibility Evaluation Tools. Disponível em: http://www.w3.org/WAI/ER/tools/complete. Informação capturada em 20/jul/2009.

WAI (2009b) W3C Web Content Accessibility Guidelines 1.0 Conformance Logos. Disponível em: http://www.w3.org/WAI/WCAG1-Conformance.html. Informação capturada em 18/ago/2009.

Wattenberg, T. L. (2006) Accessibility heuristics utilizing learnability characteristics of synthesized speech applications. In ACM SIGACCESS Accessibility and Computing, 84 (Jan. 2006), 45-47.

WCAG Samurai (2008) Introduction to WCAG Samurai errata for Web Content Accessibility Guidelines (WCAG) 1.0. Disponível em: http://wcagsamurai.org/errata/intro.html. Informação capturada em 20/ago/2009.

WebAIM (2009) Cognitive Disabilities. Disponível em: http://www.webaim.org/articles/cognitive/. Informação capturada em 29/dez/2009.

Wendt, E. (2001) Eutanásia. In: Datavenia no. 51, UEPB, out/2001. Disponível em: http://www.datavenia.net/artigos/Teoria_Geral/EUTANASIA.htm. Informação capturada em 20/ago/2009.

Wikipedia (2009a) Heuristic. Disponível em: http://en.wikipedia.org/wiki/Heuristic. Informação capturada em 20/jul/2009.

Wikipedia (2009b) Guideline. Disponível em: http://en.wikipedia.org/wiki/Guideline. Informação capturada em 20/jul/2009.

Zajicek, M.; Edwards, A. (2004) Universal usability revisited. In Interacting with Computers, Volume 16, Issue 3, Universal Usability Revisited, June 2004, Pages 403-410.

Anexo A

Materiais do experimento com ferramentas de acessibilidade

As próximas páginas apresentam todo o material utilizado no experimento com ferramentas de acessibilidade, descritos no Capítulo 3. São eles:

- **Termo de consentimento**, tratando das questões éticas envolvidas e a garantia do anonimato do participante
- **Questionário inicial**, para o levantamento do conhecimento dos participantes com relação à acessibilidade, aplicado a todos os participantes antes de iniciar qualquer teste
- **Questionário de avaliação de ferramenta de acessibilidade**, aplicado assim que o participante finalizava o teste de uma ferramenta de acessibilidade, para mensurar sua satisfação com a ferramenta
- **Questionário final**, para o levantamento das preferências dos participantes pelas ferramentas e suas impressões finais sobre as WCAG 1.0 e o experimento

Termo de consentimento

O estudo em questão pretende coletar, de forma empírica, dados a respeito da usabilidade de ferramentas de auxílio à avaliação de acessibilidade Web, bem como a respeito das *guidelines* nas quais tais ferramentas se baseiam, sendo conduzido pelo aluno de doutorado Eduardo Hideki Tanaka sob orientação da professora Dra. Heloísa Vieira da Rocha, ambos do Instituto de Computação (IC) da Universidade Estadual de Campinas (Unicamp).

Todos os dados coletados serão utilizados única e exclusivamente para tal estudo, sendo confidenciais. O meu nome e informações pessoais que possibilitariam a minha identificação, como endereço e telefone, serão mantidos em sigilo.

O estudo será realizado em locais e horários a serem definidos de acordo com as minhas preferências e disponibilidades, tendo eu total liberdade de desistir a qualquer momento de participar do mesmo.

Finalmente, declaro, na presente data, ter idade maior ou igual a 18 anos e concordar com os termos dispostos nesse documento.

Responsáveis pelo estudo:
Eduardo Hideki Tanaka, Heloísa Vieira da Rocha – IC/UNICAMP
Av. Albert Einstein, 1251 – Campinas, SP
{etanaka, heloisa}@ic.unicamp.br
Fones: (19) 3521-03431, 3521-58662

Nome do participante: _____

Assinatura do participante: _____

Local e data: _____, ____ de _____ de _____

Questionário inicial

Para uso do experimentador (não preencher):

Usuário ID: _____
Data: _____

1. Usando suas próprias palavras, o que é acessibilidade?
_____
_____
_____

Para as próximas questões, assinale com um "X" a alternativa que considerar mais apropriada.

2. Conheço pelo menos um método de inspeção de usabilidade em sistemas de informação.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

3. Utilizo frequentemente métodos de inspeção de usabilidade em sistemas de informação.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

4. O tema acessibilidade em sistemas de informação foi tratado de forma ampla em minha graduação e/ou pós-graduação.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

5. Sinto-me confiante em aplicar na prática conceitos e técnicas de design acessível de sistemas de informação.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

6. Nos projetos nos quais tenho/tive participação até hoje, questões relacionadas com acessibilidade são/foram tratadas de alguma forma.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

7. Conheço o conjunto de *guidelines* de acessibilidade Web da W3C.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

8. Conheço ferramentas de verificação de conformidade com as *guidelines* da W3C.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

   a. Cite as ferramentas que conhece (se alguma).
   _____
   _____
   _____
   _____

Questionário de avaliação de ferramenta de acessibilidade

Para uso do experimentador (não preencher):

Usuário ID: _____
Ferramenta: _____
Data: _____

Por favor, avalie a ferramenta testada marcando com um "X" a opção que considerar mais adequada para cada uma das questões a seguir.

1. A ferramenta facilita localizar os problemas de acessibilidade nas páginas Web.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

2. As justificativas para cada problema de acessibilidade ajudam a entender o porquê e quais usuários teriam dificuldades ou seriam impedidos de usar as páginas Web.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

3. A apresentação de tais justificativas é feita de forma clara e objetiva.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

4. Os relatórios da ferramenta contribuem para identificar e classificar os problemas de acessibilidade de acordo com sua gravidade.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

5. Os relatórios da ferramenta indicam possíveis formas de se corrigir os problemas de acessibilidade.
    ( ) Discordo totalmente
    ( ) Discordo parcialmente
    ( ) Não concordo nem discordo
    ( ) Concordo parcialmente
    ( ) Concordo totalmente

6. A ferramenta me passa confiança; acredito que corrigindo todos os problemas apresentados por ela estaria melhorando a acessibilidade das páginas Web analisadas.
    ( ) Discordo totalmente
    ( ) Discordo parcialmente
    ( ) Não concordo nem discordo
    ( ) Concordo parcialmente
    ( ) Concordo totalmente

Comentários adicionais sobre a ferramenta (opcional):
_____
_____
_____
_____
_____

Questionário final

Para uso do experimentador (não preencher):

Usuário ID: _____
Data: _____

Por favor, atribua notas para as ferramentas testadas com relação aos seguintes critérios (marque com um "X" a coluna com a nota desejada para cada ferramenta):

1. Apresentação dos resultados/relatórios de forma clara e objetiva:

	1. péssimo	2. ruim	3. regular	4. bom	5. excelente
HTML Validator para o Firefox	☐	☐	☐	☐	☐
Truwex	☐	☐	☐	☐	☐
daSilva	☐	☐	☐	☐	☐
Fujitsu Web Accessibility Inspector	☐	☐	☐	☐	☐

2. Justificativas apresentadas para os problemas:

	1. péssimo	2. ruim	3. regular	4. bom	5. excelente
HTML Validator para o Firefox	☐	☐	☐	☐	☐
Truwex	☐	☐	☐	☐	☐
daSilva	☐	☐	☐	☐	☐
Fujitsu Web Accessibility Inspector	☐	☐	☐	☐	☐

3. Facilidade em localizar os problemas na página Web:

	1. péssimo	2. ruim	3. regular	4. bom	5. excelente
HTML Validator para o Firefox	☐	☐	☐	☐	☐
Truwex	☐	☐	☐	☐	☐
daSilva	☐	☐	☐	☐	☐
Fujitsu Web Accessibility Inspector	☐	☐	☐	☐	☐

4. Confiança e credibilidade dos resultados/relatórios:

	1. péssimo	2. ruim	3. regular	4. bom	5. excelente
HTML Validator para o Firefox	☐	☐	☐	☐	☐
Truwex	☐	☐	☐	☐	☐
daSilva	☐	☐	☐	☐	☐
Fujitsu Web Accessibility Inspector	☐	☐	☐	☐	☐

Assinale com um “X” a opção que considerar mais adequada para cada uma das questões a seguir.

5. A experiência com as ferramentas e as *guidelines* contribuiu para melhorar meu entendimento sobre acessibilidade.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

6. Considero que o tempo necessário para verificar as páginas Web para os três níveis de prioridade das *guidelines* é adequado, independente da ferramenta.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

7. A organização das *guidelines* da forma como é proposta pela W3C é bem simples, facilitando a navegação e a consulta durante a avaliação.
   ( ) Discordo totalmente
   ( ) Discordo parcialmente
   ( ) Não concordo nem discordo
   ( ) Concordo parcialmente
   ( ) Concordo totalmente

8. Você gostaria que houvesse algo mais nas ferramentas e/ou nas *guidelines* da W3C (funcionalidade, documentação, etc.)? O que?
   _____
   _____
   _____
   _____
   _____

Comentários finais sobre os testes (opcional):
_____
_____
_____
_____
_____

Anexo B

Interfaces exploradas na fase de extração

A fim de encontrar o conjunto inicial de heurísticas de acessibilidade durante a fase de extração apresentada no Capítulo 4, diversas interfaces foram exploradas, procurando-se por potenciais problemas de acessibilidade, justificados a partir do meu conhecimento externalizado.

A seguir, são apresentadas brevemente as interfaces que foram exploradas, juntamente com os problemas de acessibilidade encontrados. A menos que não seja explicitamente indicado, as interfaces foram exploradas em um computador com Windows XP e navegador Firefox com a extensão FireVox[24] (um leitor de telas) instalada.

B.1. Microscópio Virtual

Trata-se de um software que está sendo desenvolvido dentro do projeto de Conteúdos Digitais do Ministério da Educação (Condigitais, 2009) pela equipe do Instituto de Biologia da UNICAMP. O usuário tem à sua disposição diversas lâminas e pode colocá-las no Microscópio Virtual para vê-las ampliadas. A Figura B.1 apresenta um exemplo de análise de lâmina.

[24] http://firevox.clcworld.net/. Informação capturada em 11/nov/2009.

Figura B. 1. Lâmina sendo analisada no Microscópio Virtual.

Problemas encontrados/comentários dos avaliadores:

- Usando o navegador Firefox, o primeiro acesso ao Microscópio Virtual usando apenas o teclado não funciona, é preciso que o usuário clique com o mouse em cima do mesmo para que o teclado passe a funcionar apropriadamente. Provavelmente, o problema ocorre pois o foco ao carregar a página não é no elemento Flash que compõe o Microscópio Virtual.
- Sempre que se carrega a página do Microscópio Virtual, o foco para navegar com o teclado já deveria estar na aplicação, independente do navegador usado, evitando que o usuário tenha que passar por vários elementos até chegar à aplicação em si.
- O Microscópio Virtual apresenta muitos textos em português escrito, o que pode ser uma dificuldade para usuários que não o dominam plenamente, como o caso de surdos que têm preferência pela língua de sinais. Uma alternativa seria o uso maior de elementos visuais/icônicos na interface.
- O Microscópio Virtual apresenta funcionalidade para aumentar/diminuir o tamanho do texto na tela. Mas a mesma não está disponível em todas as telas e, mesmo nas telas em que ele está disponível, nem todos os textos (rótulos, mensagens, títulos, etc.) alteram de tamanho.

- Quando estas e outras funcionalidades de acessibilidade são oferecidas pelo software, deveriam ser facilmente reconhecidas, ligadas, desligadas e configuráveis por qualquer usuário que efetivamente precise utilizá-las.
- O conteúdo do Microscópio deveria ser condizente a todos os públicos. Por exemplo, para um cego, a mensagem "observe os núcleos das células..." não é adequada; seria um pouco mais apropriado se houvesse uma descrição da imagem mostrada no microscópio (formato da célula, tamanho dos núcleos, etc.). Da mesma maneira, para um usuário surdo acessando uma música ou podcast, poderia-se disponibilizar também a letra ou transcrição.
- Ao fornecer teclas de atalho para as principais funcionalidades, a interface acaba se tornando mais eficiente também para cegos que necessitam navegar usando apenas o teclado.
- Para facilitar a navegação, a informação sobre o histórico de páginas/telas navegadas do Microscópio Virtual poderia ser mostrada ao usuário, que muitas vezes tem ir e voltar várias vezes (para a tela de escolha de lâminas, para o microscópio em si...), tal como vários sites já utilizam.

B.2. Quebra-Cabeça

É um programa disponível no Portal do Professor (MEC, 2009), cujo objetivo é colocar os Estados brasileiros no seu respectivo local no mapa do Brasil, conforme pode ser visto na Figura B.2. Para cada acerto, o jogador ganha um ponto e, para cada erro, perde um.

Figura B. 2. Jogo do Quebra-Cabeça, com parte dos Estados fora de posição.

Problemas encontrados/comentários dos avaliadores:

- O jogo do Quebra-Cabeça só funciona com o mouse, através da ação de arrastar e soltar, sendo impossível o uso do mesmo apenas com o teclado – o que impede seu uso por pessoas com deficiências motoras, por exemplo. Também, seria praticamente impossível usá-lo com leitores de tela, uma vez que, em geral, não se usa o mouse ou outros dispositivos de apontamento com tais tecnologias assistivas.

B.3. Jogo da Reciclagem

Outro programa do Portal do Professor (MEC, 2009), também baseado em interações do tipo arrastar e soltar. No Jogo da Reciclagem, o usuário é apresentado a um objeto que deverá ser colocado em uma lixeira específica para metal, vidro, plástico ou papel – as lixeiras são identificadas apenas por cores. Na Figura B.3, pode-se observar uma tela do Jogo da Reciclagem, na qual o usuário deve selecionar qual é a lixeira adequada para descartar uma seringa.

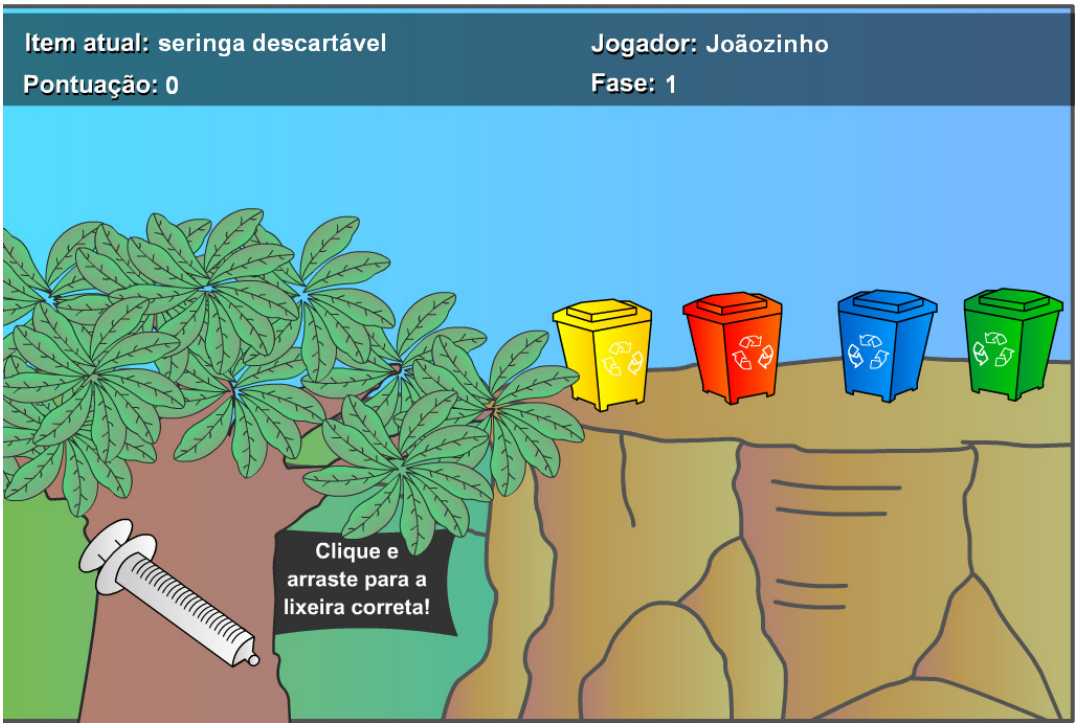


Figura B. 3. Escolha de lixeira no Jogo da Reciclagem.

Problemas encontrados/comentários dos avaliadores:

- O jogo possui elementos que passam pela tela, por exemplo, uma mensagem a cada nova fase, mas o usuário não tem como configurar a velocidade dos mesmos de acordo com sua necessidade e habilidade em ler. Sendo assim, o jogo poderia fornecer controles para pausar/continuar, reiniciar e rever a mensagem de nova fase, por exemplo.
- Apesar da simplicidade, só é possível interagir através do mouse. Teclas de atalho ou mesmo possibilitar o uso das setas do teclado para levar o objeto sendo mostrado direto para uma das lixeiras seriam soluções muito bem vindas para usuários que não conseguem usar o mouse.
- As lixeiras são diferenciadas apenas através da cor, fazendo com que a identificação de cada uma possa ser difícil por pessoas com cromodeficiências ou monitores monocromáticos. Na vida real, sempre há rótulos em lixeiras de coleta de lixo reciclável, mas no jogo não.

- No final do jogo, todas as faixas de pontuação possíveis são apresentadas ao usuário, com um destaque na cor do fundo do texto da faixa que o usuário alcançou. Mostrar todas as faixas de pontuação pode ser relevante para o usuário comparar os resultados possíveis, mas, ao mesmo tempo, sobrecarrega a tela com muito texto. Só se deveriam apresentar informações relevantes, para não sobrecarregar o usuário e para que o uso com leitor de telas seja mais eficiente – o usuário não precisará ouvir muitas frases até encontrar aquela que é mais relevante ou importante para ele no momento.

B.4. Cada Caso é um Caso

O Cada Caso é um Caso é um software distribuído pela RIVED, a Rede Interativa Virtual de Educação (RIVED, 2009). No Cada Caso é um Caso, são apresentados vários casos de pessoas com alguns sintomas, como dor de cabeça, febre, entre outros. Baseado nessas informações e nas descrições de classes terapêuticas presentes, o usuário deve selecionar um medicamento apropriado, como pode ser visto em no exemplo da Figura B.4.

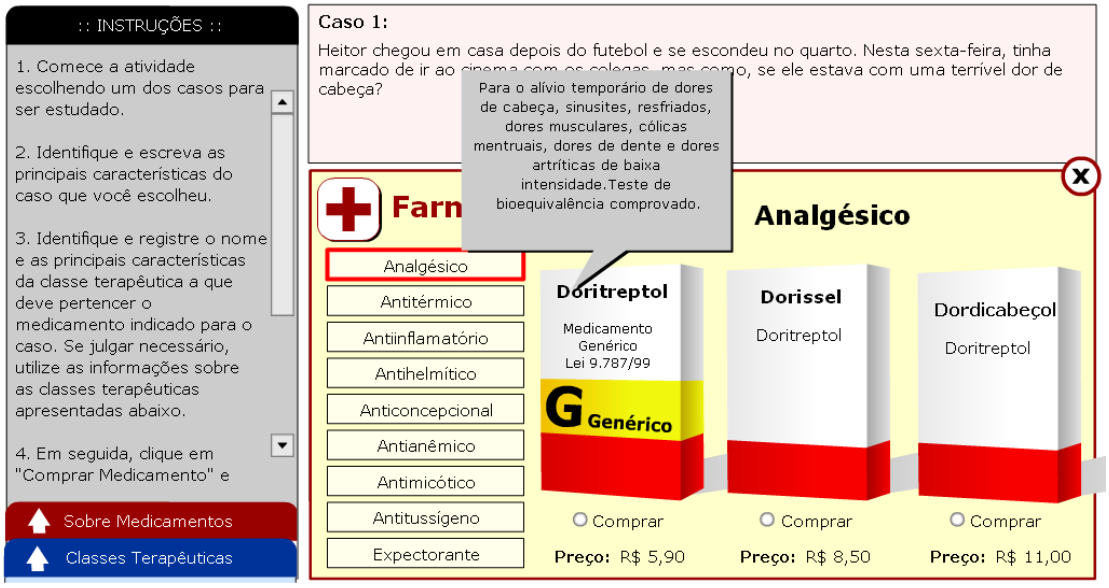


Figura B. 4. Escolha de analgésico no Cada Caso é um Caso.

Problemas encontrados/comentários dos avaliadores:

- A interface é estritamente textual e requer que o usuário seja fluente em Português, o que pode não ser o caso de todos os surdos, por exemplo, uma vez que muitos preferem usar a língua de sinais ao Português. Assim, poderia haver sinalização como uma alternativa.
- A navegação usando apenas a tecla "tab" do teclado é em uma ordem aleatória e não muito lógica entre os elementos da tela, dificultando a localização do controle na tela.
- O uso de elementos escondidos, como textos em páginas Web que têm a mesma cor do fundo para que fiquem "invisíveis" para usuários videntes, deve ser feita com cuidado, uma vez que elementos assim podem ser lidos por leitores de telas. Ou seja, no mínimo, incluir este tipo de elemento irá aumentar a quantidade de texto a ser lido pelo leitor de telas e potencialmente reduzir a eficiência de uso da interface com este tipo de tecnologia assistiva.

B.5. Geometria

Trata-se de outro programa da RIVED (2009), no qual o usuário tem que identificar formas geométricas e pode analisar, para cada uma delas, suas faces, vértices e arestas. O usuário também pode classificar as formas geométricas que observou e, depois realizar uma atividade para relacioná-las com as que aparecem em fotos de cidades reais. A Figura B.5 ilustra a análise de um poliedro no programa Geometria.

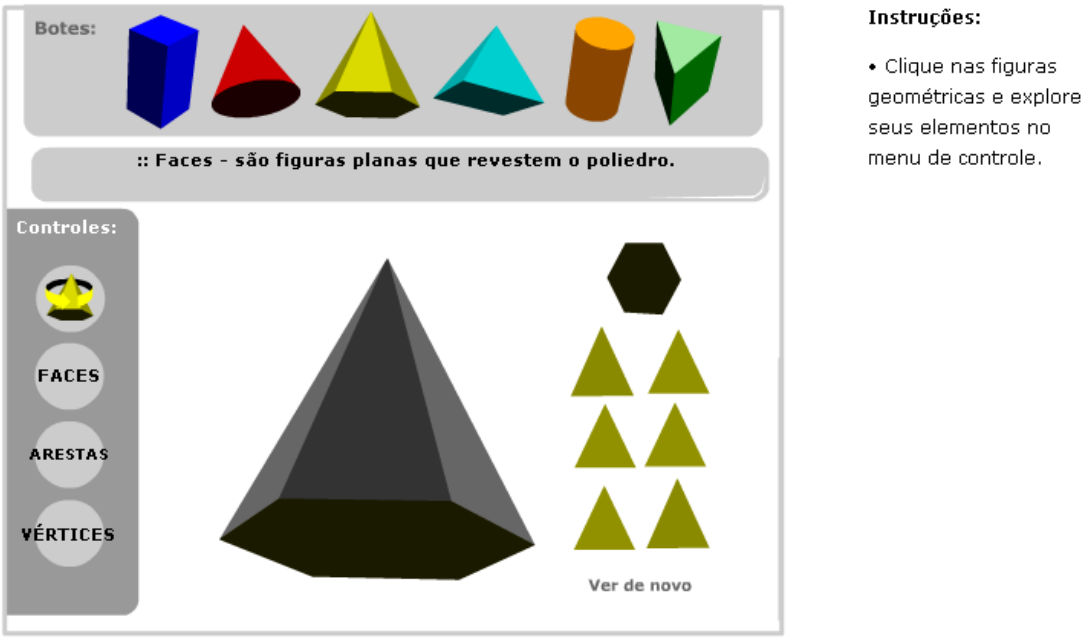


Figura B. 5. Análise de um poliedro no software Geometria.

Problemas encontrados/comentários dos avaliadores:

- A interação com o software só pode ser feita através do mouse. Caso o usuário tente utilizar o teclado para interagir, a solução é automaticamente selecionada, acabando com o objetivo do jogo.
- O software é extremamente visual e alternativa textuais não parecem ser muito adequadas. A alternativa mais apropriada, neste caso, seria o uso de interfaces físicas/táteis, que possibilitam o usuário cego a analisar as formas. Entretanto, interfaces deste tipo ainda não estão muito disseminadas.
- O Geometria ignora as configurações de cores definidas no sistema operacional. Assim, usuários que precisam usar esquemas de cores em alto contraste poderão sentir dificuldades ao interagir com o software.

B.6. Não me sinto bem

Outro software da RIVED (2009), o Não me sinto bem é um jogo para duas pessoas, uma controla o vilão e a outra o mocinho. Para cada doença que é apresentada, o vilão deve escolher algo que a potencialize ou uma forma como ela é contraída ao passo que o mocinho deve selecionar algo que a evita. Por exemplo, no caso da doença de Chagas, um vilão poderia escolher a alternativa "casas de pau a pique" para ganhar pontos enquanto um mocinho poderia escolher "erradicação do inseto barbeiro". No exemplo da Figura B.6 é possível observar algumas opções do jogo quando a doença apresentada é a Malária.

Figura B. 6. Não me sinto bem, quando a doença mostrada é a malária.

Problemas encontrados/comentários dos avaliadores:

- Existe um tempo para que cada jogador escolha uma opção na tela, mas um usuário com leitor de telas nunca terá esta informação.
- O tempo para escolha de uma das opções disponíveis não é configurável de acordo com as preferências e necessidades do usuário. Por exemplo, um usuário surdo pode não ser tão fluente no Português e pode precisar de mais tempo para ler cada uma das opções na tela.

B.7. GeoNet

O GeoNet (Eduplace, 2009ª) é um jogo do tipo perguntas e respostas relacionadas à geografia dos Estados Unidos ou do resto do mundo. À medida que vai acertando, o jogador vai ganhando pontos e melhorando seu "status". É um dos jogos encontrados na lista de programas acessíveis em Flash da própria Adobe[25]. Na Figura B.7, é possível ver uma pergunta específica sobre o tamanho dos países da América do Sul no GeoNet.

Figura B. 7. Perguntas sobre os países da América do Sul no GeoNet.

[25] http://www.adobe.com/accessibility/examples.html. Informação capturada em 11/nov/2009.

Problemas encontrados/comentários dos avaliadores:

- Para melhorar o feedback para quem usa leitor de telas, poder-se-ia incluir uma barra de progresso com o tempo estimado para terminar de carregar o software.
- A cor do fundo em algumas telas é muito parecida com a cor de seleção/foco de elementos do Flash, o que pode trazer algumas dificuldades aos usuários em diferenciá-los.
- A ajuda está disponível em língua escrita apenas. Poderiam ser incluídas figuras, animações e mesmo sinalização para facilitar a sua compreensão por usuários não tão fluentes na língua escrita.
- O software não permite que o usuário dê zoom pelo navegador. Assim, usuários com visão subnormal podem não conseguir visualizar de forma apropriada o conteúdo completo do GeoNet.

B.8. Pre-Season Puzzles

O Pre-Season Puzzles (Eduplace, 2009b) é um jogo de palavras cruzadas relacionado com futebol americano. Assim como o GeoNet, também consta da lista de programas acessíveis em Flash da própria Adobe. Na Figura B.8, pode-se observar as palavras cruzadas do Pre-Season Puzzles.

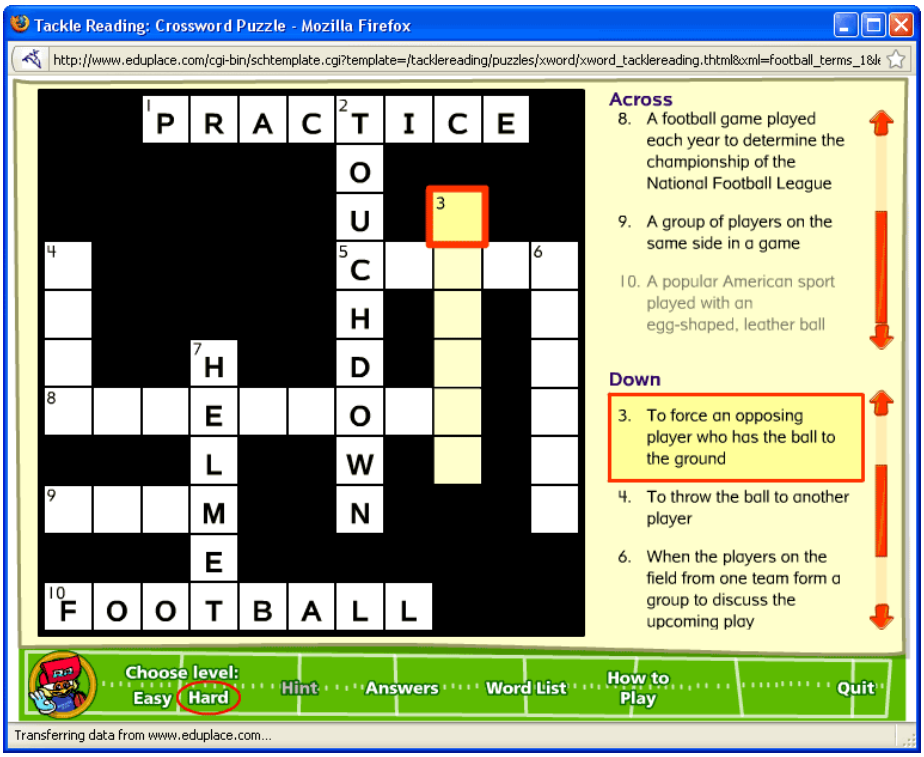


Figura B. 8. Palavras cruzadas no Pre-Season Puzzles.

Problemas encontrados/comentários dos avaliadores:

- Em "How to play", na primeira tela há um link "previous" que, ao ser lido no leitor de telas, será apenas mais uma informação irrelevante para o usuário, visto que o mesmo está desabilitado e não há uma página anterior pois já se está na primeira página.
- Como no Geonet, a ajuda é apenas textual.
- Dependendo da fonte que o usuário escolhe para usar, ao aumentar a tela (zoom) pelo navegador algumas letras ficam borradas e difíceis de ler.
- A cor da linha/coluna selecionada é muito parecida com a das demais, dificultando a sua identificação na tela, especialmente por usuários com deficiências visuais ligadas à identificação de cores.
- Usando apenas o teclado, é praticamente impossível acessar algumas opções disponíveis na parte de baixo da tela, como "Hint" e "Word list" – elas não são acessíveis pelo "tab" muito menos pelas setas do teclado. Os desenvolvedores

criaram atalhos para estas opções que exigem o uso do “Caps Lock”, totalmente fora dos padrões de atalhos e colocando uma funcionalidade incomum para o “Caps Lock”.

B.9. Pidgin

O Pidgin (2009) é um programa de mensagens instantâneas compatível com diversos protocolos, dentre os quais MSN, ICQ, Yahoo, entre outros. Há versão para GNU/Linux e Windows, sendo que a versão utilizada era para GNU/Linux. A Figura B.9 mostra a tela de troca de mensagens do Pidgin.

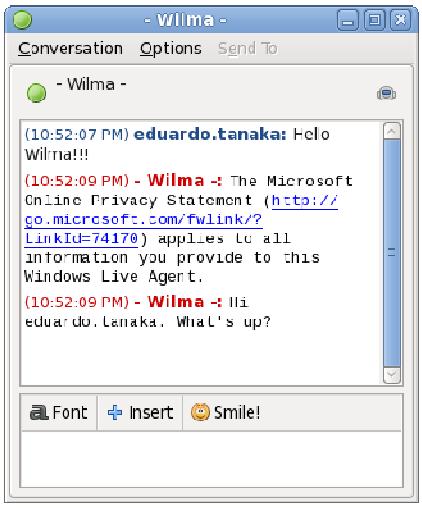


Figura B. 9. Troca de mensagens no Pidgin.

Problemas encontrados/comentários dos avaliadores:

- Na tela de configuração do Pidgin, a seleção do protocolo a ser usado não é acessível pelo teclado, apenas pelo mouse.
- A diferença dos estados dos contatos na lista de contatos se dá pela cor de um círculo ao lado de cada contato. Entretanto, as cores para os estados “idle” e “online” são muito parecidas (tons de verde), o que pode confundir alguns usuários.

- Outro problema com os estados é que não há uma legenda para identificá-los.
- Algumas informações dos contatos (se estão bloqueados, se possuem você na lista de contatos, etc.) são visíveis apenas quando o usuário coloca o cursor do mouse em cima da sua lista de contatos, ou seja, usuários que só usam teclado não têm acesso a estas informações.
- Com o leitor de telas, é impossível saber se alguém está digitando uma resposta, apesar de aparecer na tela de troca de mensagens um alerta "Usuário está digitando...". Ou seja, não há um feedback equivalente e total do que efetivamente aparece na tela para o leitor de telas.
- Ainda com o leitor de telas, na lista de contatos, o estado de cada contato não é lido, a menos que a opção de mostrar detalhes esteja habilitada.
- Outro problema é que o leitor de telas não lê os "smiles" que são recebidos nas mensagens.

Anexo C

Materiais usados na fase de refinamento

As próximas páginas apresentam os materiais utilizados na fase de refinamento descrita no Capítulo 4:

- Termo de consentimento, tratando das questões éticas envolvidas e a garantia do anonimato do participante
- Exemplos de violações das heurísticas de acessibilidade, que foi impresso e distribuído aos participantes dos experimentos durante a fase de refinamento para auxiliá-los nas avaliações de acessibilidade usando as heurísticas propostas
- Questionários sobre a avaliação heurística de acessibilidade, para aferir a satisfação dos avaliadores com as heurísticas de acessibilidade propostas, do primeiro e do segundo experimentos

Termo de consentimento

O estudo em questão pretende coletar, de forma empírica, dados a respeito de um conjunto de heurísticas de acessibilidade proposto pelo aluno de doutorado Eduardo Hideki Tanaka, sob orientação da professora Dra. Heloísa Vieira da Rocha, ambos do Instituto de Computação (IC) da Universidade Estadual de Campinas (Unicamp).

Todos os dados coletados serão utilizados única e exclusivamente para tal estudo, sendo confidenciais. O meu nome e informações pessoais que possibilitam a minha identificação, como endereço e telefone, serão mantidos em sigilo.

O estudo será realizado em locais e horários a serem definidos de acordo com as minhas preferências e disponibilidades, tendo eu total liberdade de desistir a qualquer momento de participar do mesmo.

Finalmente, declaro, na presente data, ter idade maior ou igual a 18 anos e concordar com os termos dispostos nesse documento.

Responsáveis pelo estudo:
Eduardo Hideki Tanaka, Heloísa Vieira da Rocha
IC/Unicamp
Av. Albert Einstein, 1251 – Campinas, SP
{etanaka, heloisa}@ic.unicamp.br
Fones: (19) 3521-03431, 3521-58662

Nome do participante: _____

Assinatura do participante: _____

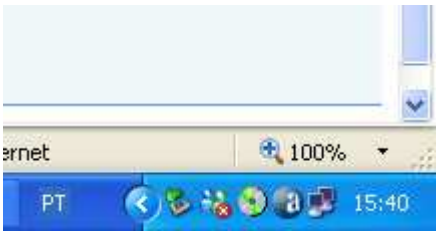
Local e data: _____, ____ de _____ de _____

Exemplos de violações das heurísticas de acessibilidade

Exemplo 1:

Ao lado do relógio do Windows XP, várias aplicações colocam símbolos e ícones para acesso rápido. Entretanto, este acesso só pode ser feito através do mouse ou outro dispositivo de apontamento, não sendo acessível pelo teclado, por exemplo. Usuários que utilizam apenas o teclado como dispositivo de entrada, portanto, não têm acesso a estes programas.

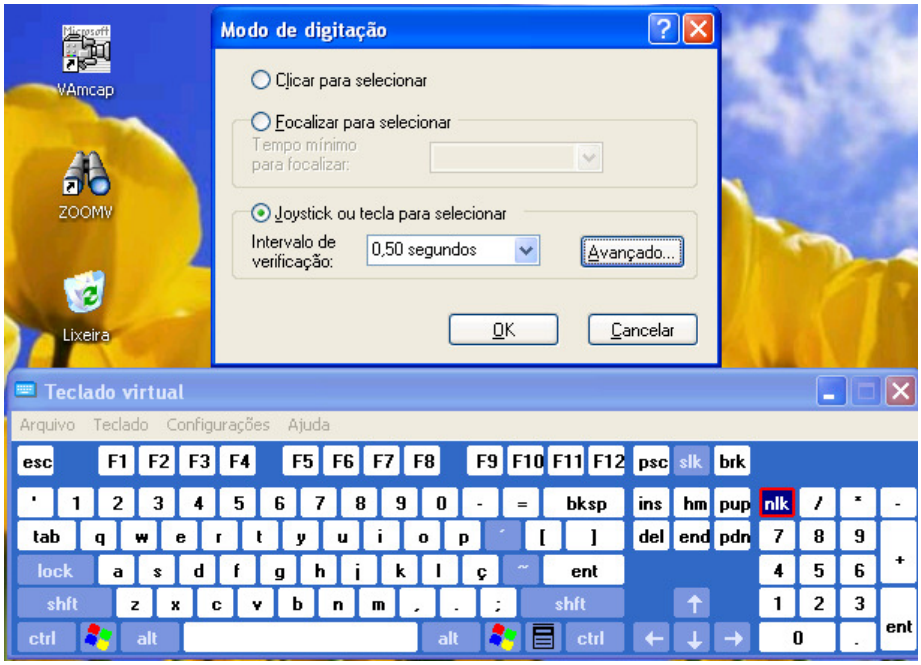
Heurística(s) violada(s): suporte a diferentes tipos de entrada e saída.



Exemplo 2:

O Windows XP apresenta vários recursos de acessibilidade, sendo um deles o Teclado Virtual. O problema é que, para ativar a varredura do Teclado Virtual, alguém precisa antes ligá-la e configurá-la, o que exige o uso do mouse e do teclado convencional. Ou seja, não é possível para o próprio usuário do Teclado Virtual do Windows XP ativá-lo inicialmente e configurá-lo de acordo com suas preferências e necessidades.

Heurística(s) violada(s): independência de uso.



Exemplo 3:

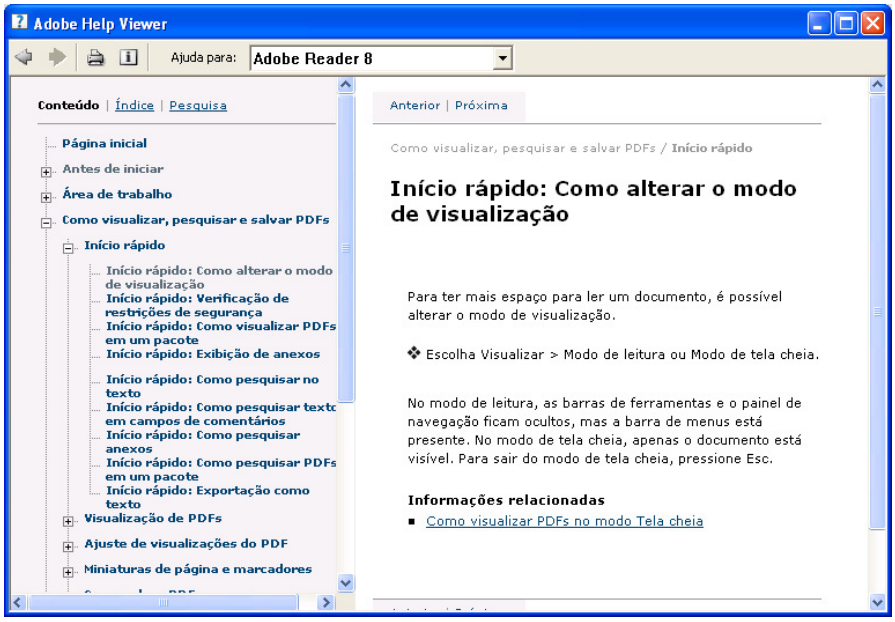
O Nero Express, programa para gravar mídias de CD e DVD, procura seguir as convenções do sistema operacional para fontes e cores, mas não em toda a sua interface. As opções de tipos de gravação, por exemplo, não se alteram quando se utilizam cores em alto contraste – e o que é pior, alguns itens de menu simplesmente desaparecem quando o usuário está utilizando cores de alto contraste no Windows.

Heurística(s) violada(s): respeito às preferências do usuário.

Exemplo 4:

A Adobe vem incluindo várias ferramentas para melhorar a acessibilidade em seus produtos. Um exemplo é a versão 8 do Adobe Reader, que já vem com uma lupa para aumentar áreas específicas da tela e um leitor de telas integrado. Entretanto, a ajuda do Acrobat Reader, apesar de ser dividida em vários documentos, é estritamente e totalmente em língua escrita, restringindo seu uso a pessoas fluentes em tal língua – surdos sem domínio da língua escrita, por exemplo, poderão ter dificuldades no entendimento da ajuda. Além disso, os recursos de acessibilidade existentes para ler documentos do tipo PDF não estão disponíveis na ajuda.

Heurística(s) violada(s): conteúdo para todos os usuários.



Exemplo 5:

O Adobe Reader 8 possui um "Assistente de configuração para acessibilidade", no qual diversas opções para leitores de tela e lupas podem ser configuradas, separadas em páginas. Contudo, o Assistente só está disponível através do menu "Documentos" e não há atalhos para acessar tais opções diretamente. Por exemplo, se o usuário desejar ativar o uso de cores de alto contraste, deverá percorrer ao menos 2 páginas do Assistente para encontrar esta opção. O usuário, então, se vê obrigado ou a utilizar o mouse ou a acessar um caminho um tanto quanto longo usando o teclado – ao todo, são pelo menos 3 passos para abrir a primeira tela do Assistente.

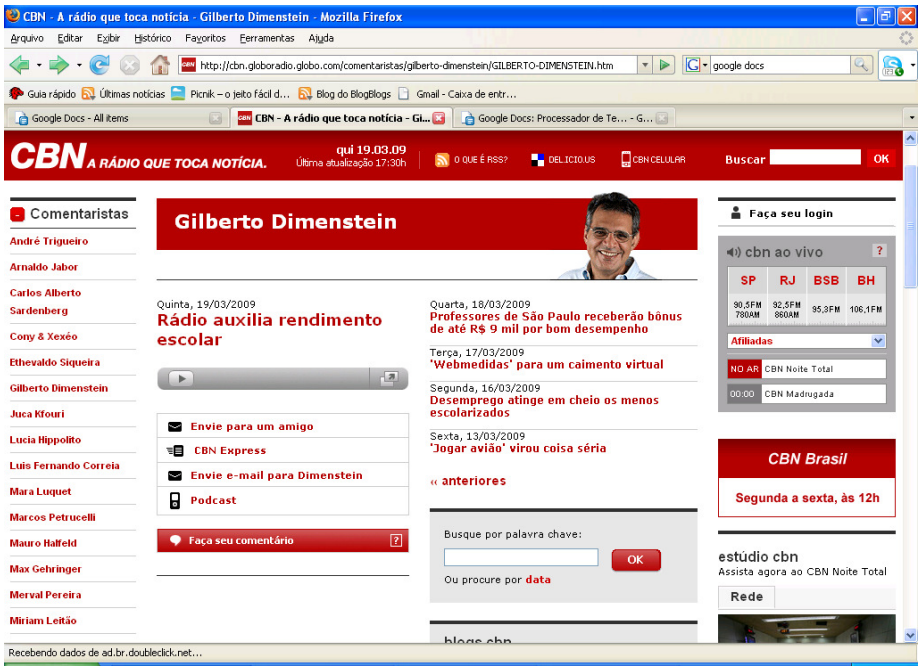
Heurística(s) violada(s): eficiência em navegação alternativa, independência de uso.

Exemplo 6:

O Podcast é como um blog em áudio que vem se tornando muito popular nos dias de hoje. Os comentaristas da Rádio CBN, por exemplo, disponibilizam Podcasts. Apesar das inovações, Podcasts como os da Rádio CBN são acessíveis apenas a usuários ouvintes –

para não falar em fluentes na língua do Podcast, já que não fornecem um equivalente alternativo aos não ouvintes ou não fluentes no idioma em que foram gravados.

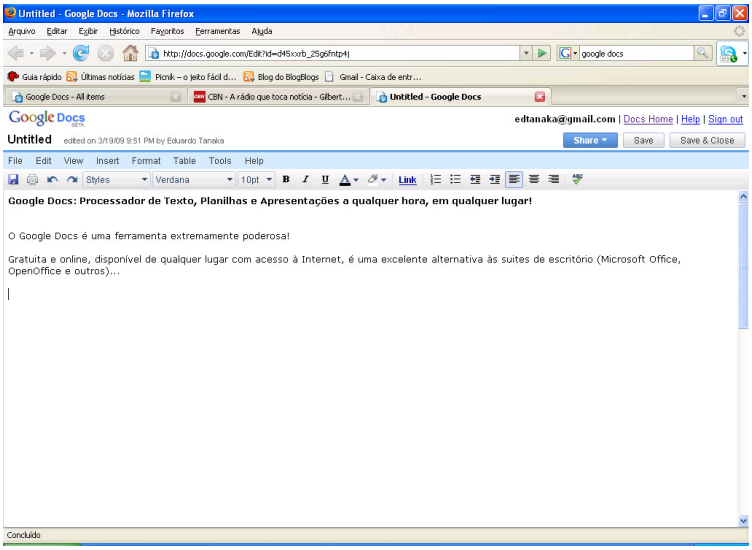
Heurística(s) violada(s): conteúdo para todos os usuários.



Exemplo 7:

O Google Docs possui praticamente as mesmas funcionalidades de uma suíte de escritório tradicional como o Microsoft Office ou o OpenOffice. Entretanto, acessar os menus de opções não é uma tarefa trivial. Ao contrário de seus similares offline, em que basta pressionar a tecla "Alt" para acessar o menu, no Google Docs é preciso clicar "Shift + Tab" duas vezes para se chegar aos menus. Ou seja, além de necessitar segurar uma tecla a mais, a operação deve ser feita duas vezes, comprometendo a eficiência de quem só consegue utilizar o computador através do teclado, sem o mouse.

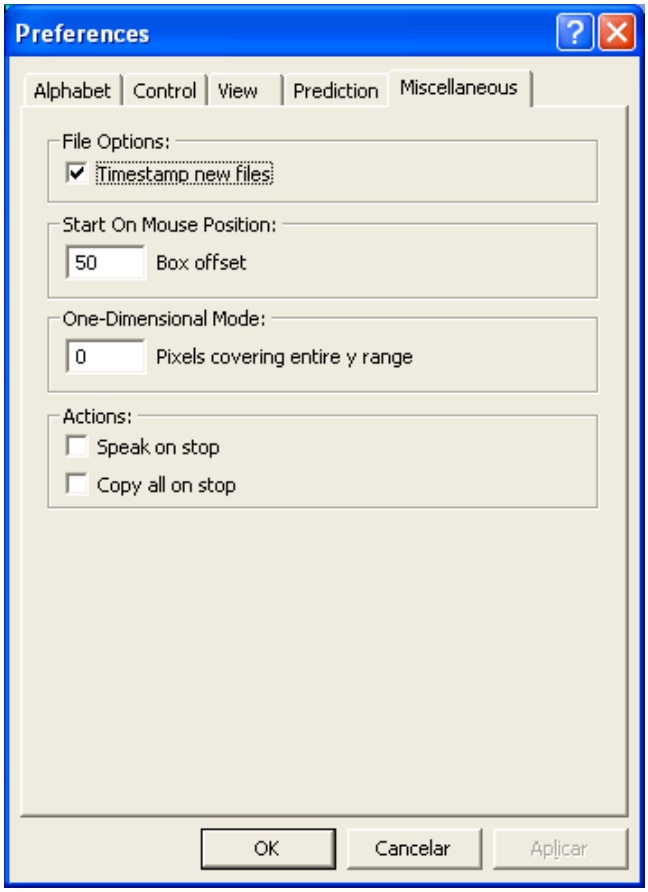
Heurística(s) violada(s): eficiência em navegação alternativa.



Exemplo 8:

O Dasher é um programa que permite a escrita usando apenas um dispositivo de apontamento, como um *eye tracker*. Apesar de ser possível abrir todas as suas janelas com o uso de um dispositivo de apontamento, nem todas as opções de preferências do usuário são facilmente alteradas com este tipo de dispositivo. Por exemplo, a posição inicial do cursor do mouse só é ajustável se o usuário digitar usando um teclado. Ou seja, o usuário que só acessa o computador através de um dispositivo de apontamento não conseguirá configurar plenamente o Dasher, pois irá precisar usar o teclado para tal ou pedir a ajuda de outro usuário que consiga usar o teclado.

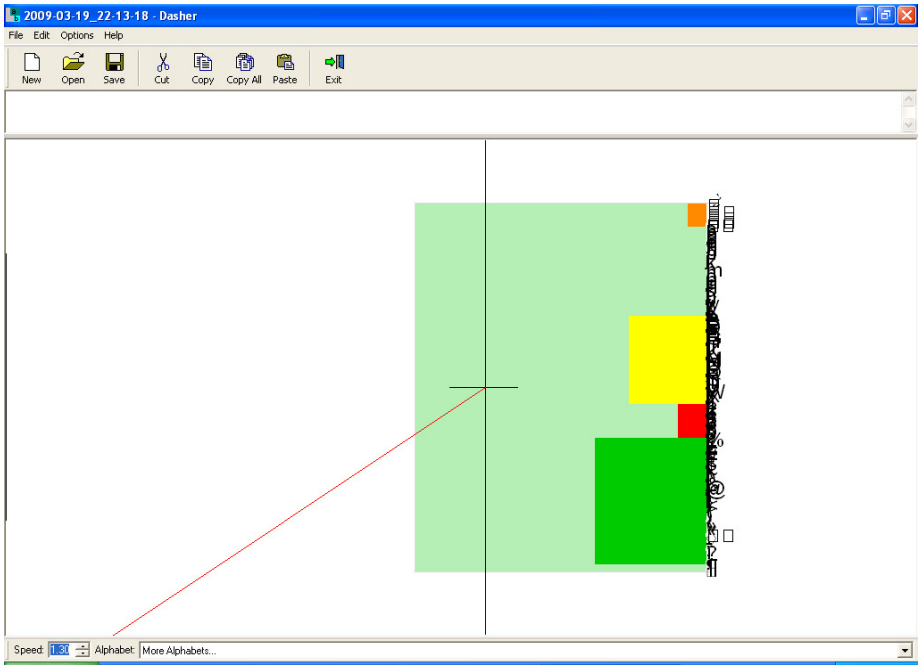
Heurística(s) violada(s): independência de uso.



Exemplo 9:

O Dasher é um software para uso com dispositivos de apontamento. O usuário pode selecionar a velocidade de varredura das letras através da caixa numérica com botões para aumentar ou diminuir a velocidade. Porém, o incremento da velocidade é sempre de um em um décimo, sendo muito pouco eficiente e podendo demandar do usuário muito tempo para configurar a velocidade mais adequada a ele(a).

Heurística(s) violada(s): eficiência em navegação alternativa.



Exemplo 10:

O FotoFlexer é um editor de imagens online com várias funções para alterar e corrigir fotos. Um problema que ocorre com o mesmo é que, por mais que o usuário tenha configurado seu navegador de Internet para utilizar um esquema de cores e um tamanho de fonte específico, o FotoFlexer permanece inalterado, desrespeitando as preferências do usuário. O usuário sequer consegue utilizar a funcionalidade do navegador para aumentar ou diminuir a fonte usada.

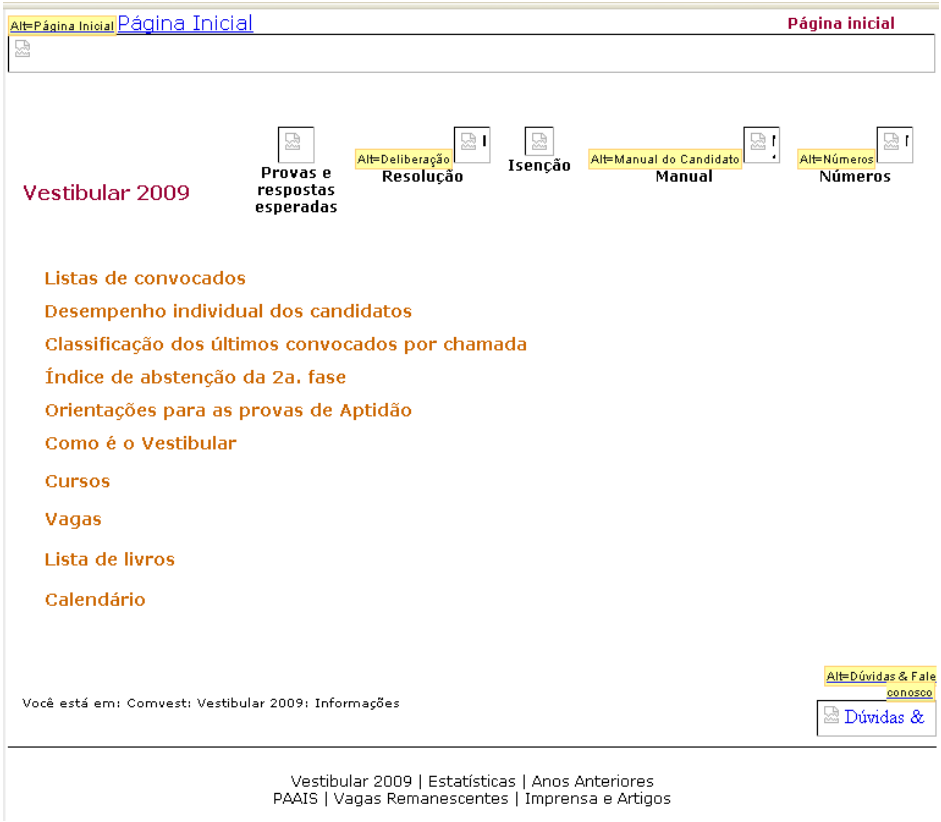
Heurística(s) violada(s): respeito às preferências do usuário.

Exemplo 11:

Muitas das figuras utilizadas no site da Comissão Permanente de Vestibulares (Comvest) da Unicamp apresentam equivalentes textuais. Contudo, como algumas destas figuras são imediatamente seguidas por rótulos explicativos, o equivalente textual se faz desnecessário e apenas aumenta o tempo de leitura por um utilizador de leitor de telas. Por exemplo, logo

na primeira linha da página do Vestibular 2009, há três ocorrências de “Página Inicial”, contando rótulos e equivalentes textuais, e os links para acessar o manual do candidato e números serão lidos duas vezes cada um.

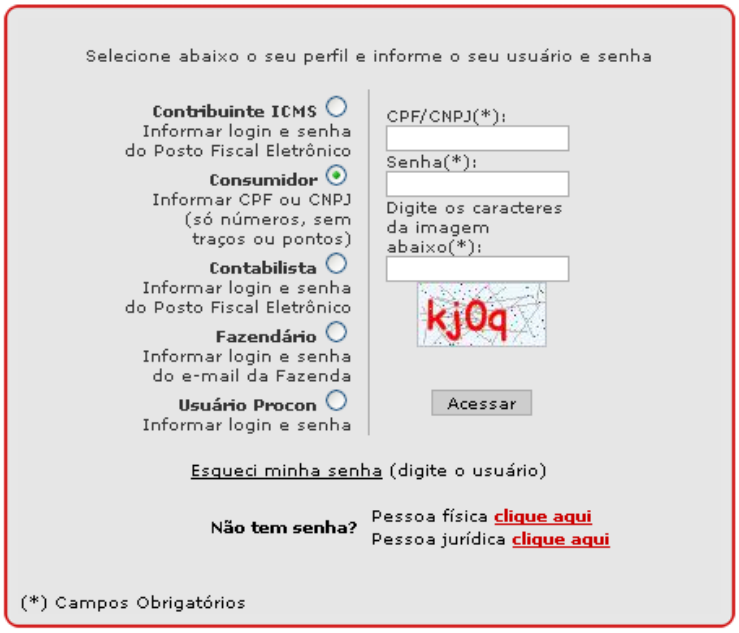
Heurística(s) violada(s): eficiência em navegação alternativa.



Exemplo 12:

Para acessar o sistema de consultas de créditos do programa “Nota Fiscal Paulista”, o usuário deve fornecer seu CPF/CNPJ, sua senha de acesso e digitar os caracteres que aparecem dentro de uma imagem. Porém, como tal imagem não apresenta um equivalente textual e o sistema não fornece outra forma de entrar na consulta de créditos, utilizadores de leitores de telas não conseguirão sequer passar desta página do programa “Nota Fiscal Paulista”.

Heurística(s) violada(s): suporte a diferentes tipos de entrada e saída.



Questionário sobre a avaliação heurística de acessibilidade

(Primeiro Experimento)

Este questionário visa coletar a sua opinião, como avaliador de interfaces, a respeito de sua experiência com as heurísticas de acessibilidade. Ressaltamos que os dados serão utilizados apenas para fins estatísticos e suas respostas serão mantidas no anonimato.
Nas questões abaixo, escolha sempre uma única opção (a que melhor expressa sua opinião).

1. O tempo para realizar uma avaliação com as heurísticas de acessibilidade é satisfatório.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

2. Aprendi rapidamente a usar as heurísticas.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

3. Tive dúvidas iniciais sobre como usar as heurísticas.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

4. As dúvidas que porventura eu tinha no início ao usar as heurísticas foram diminuindo com o progresso da avaliação e a(s) reunião(ões) para discutí-las.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

5. As heurísticas de acessibilidade são mais fáceis de aplicar do que as *guidelines* da W3C.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

6. O material de apoio cedido, contendo a descrição das heurísticas e exemplos de violações, é claro e de fácil compreensão.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

7. Eu entendi do que trata a heurística “Suporte a diferentes tipos de entrada e saída”.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

8. Eu entendi do que trata a heurística “Conteúdo para todos os usuários”.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

9. Eu entendi do que trata a heurística “Independência de uso”.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

10. Eu entendi do que trata a heurística “Respeito às preferências do usuário”.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

11. Eu entendi do que trata a heurística “Eficiência em navegação alternativa”.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

12. Encontrei problemas graves de acessibilidade usando as heurísticas.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

13. Encontrei problemas de acessibilidade que nem imaginava que existiam usando as heurísticas.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

14. Tentaria usar novamente as heurísticas se fosse necessário avaliar a acessibilidade de outro sistema.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

15. No geral, me senti satisfeito com os resultados da minha avaliação de acessibilidade.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

16. Acredito que posso aprimorar mais minha capacidade de avaliar a acessibilidade usando as heurísticas outras vezes.

( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

17. Se você pudesse alterar uma heurística, qual seria? E o que mudaria?

18. Cite duas coisas que você gostou e duas que não gostou ao usar as heurísticas de acessibilidade.

19. Caso queira, escreva comentários, críticas ou sugestões a respeito da avaliação com heurísticas de acessibilidade.

Questionário sobre a avaliação heurística de acessibilidade

(Segundo Experimento)

Este questionário visa coletar a sua opinião, como avaliador de interfaces, a respeito de sua experiência com as heurísticas de acessibilidade. Ressaltamos que os dados serão utilizados apenas para fins estatísticos e suas respostas serão mantidas no anonimato.

Nas questões abaixo, escolha sempre uma única opção (a que melhor expressa sua opinião).

1. O tempo para realizar uma avaliação com as heurísticas de acessibilidade é satisfatório.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

2. De forma geral, as heurísticas estavam descritas de forma clara e de fácil compreensão.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

3. Eu entendi do que trata a heurística "Suporte a diferentes tipos de entrada e saída".
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

4. Eu entendi do que trata a heurística "Conteúdo para todos os usuários".
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

5. Eu entendi do que trata a heurística "Independência de uso".
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

6. Eu entendi do que trata a heurística "Respeito às preferências do usuário".
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

7. Eu entendi do que trata a heurística "Eficiência em navegação alternativa".
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

8. Encontrei problemas graves de acessibilidade usando as heurísticas.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

9. Encontrei problemas de acessibilidade que nem imaginava que existiam usando as heurísticas.
   ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

10. Tentaria usar novamente as heurísticas se fosse necessário avaliar a acessibilidade de outro sistema.
    ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

11. No geral, me senti satisfeito com os resultados da minha avaliação de acessibilidade.
    ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

12. Acredito que posso aprimorar mais minha capacidade de avaliar a acessibilidade usando as heurísticas outras vezes.
    ( ) Discordo Totalmente ( ) Discordo ( ) Indiferente ( ) Concordo ( ) Concordo Totalmente

13. Caso queira, escreva comentários, críticas ou sugestões a respeito da avaliação com heurísticas de acessibilidade.

Anexo D

Resultados da avaliação de acessibilidade – primeiro experimento

A tabela a seguir apresenta os problemas de acessibilidade encontrados para o Google Docs durante o primeiro experimento da fase de refinamento das heurísticas, como descrito no Capítulo 4. Na coluna "Participantes", P1, P2, P3 e P4 representam os participantes que encontraram um determinado problema, por exemplo, o primeiro problema foi descoberto apenas por um participante, P1, ao passo que o quinto problema foi identificado por 3 participantes, P1, P2 e P3.

Tabela D. 1. Problemas encontrados durante o primeiro experimento com as heurísticas de acessibilidade.

#	**Problema**	**Heurística(s) violada(s)**	**Participantes**
1	Como o foco da página começa na caixa de edição do documento, se o usuário estiver usando o leitor de telas, ele não irá ler nada. Como o usuário sequer sabe onde ele se encontra, provavelmente terá que ir com o leitor para o início da página usando o teclado.	Eficiência em navegação alternativa	P1
2	Não é possível acessar alguns botões da barra de ferramentas usando apenas o teclado.	Suporte a diferentes tipos de entrada e saída	P1
3	Navegando com o leitor de telas, os menus do Google Docs são interpretados e lidos como "botões", o que pode confundir o usuário.	Suporte a diferentes tipos de entrada e saída, Eficiência em	P1

		navegação alternativa	
4	Os botões da barra de ferramentas não são visíveis quando o usuário configura o sistema para utilizar cores em alto contraste.	Respeito às preferências do usuário	P1, P2, P3, P4
5	No Google Docs, não há uma forma de se aumentar a fonte ou trocar a cor da interface; isto só é possível via navegador ou sistema operacional.	Conteúdo para todos os usuários	P1, P2, P3
6	Na navegação usando apenas a tecla “Tab”, não é possível acessar os menus, é preciso utilizar as setas do teclado também.	Suporte a diferentes tipos de entrada e saída, Eficiência em navegação alternativa	P1
7	Quando se insere uma tabela, o cursor vai para uma linha logo abaixo na tabela, e não para dentro da mesma. Alguém que só navega usando teclado teria que então voltar para a primeira célula da tabela usando as setas do teclado para começar a preenchê-la.	Eficiência em navegação alternativa	P1
8	Não é possível navegar pelas células de uma tabela usando a tecla “Tab” do teclado.	Eficiência em navegação alternativa	P1
9	Não há alternativa à língua de sinais para os textos.	Conteúdo para todos os usuários	P1, P2
10	O leitor de telas não lê os botões de Salvar, Desfazer e Refazer.	Suporte a diferentes tipos de entrada e saída	P2
11	Usando o leitor de telas para ler desde o início da página, após a leitura dos botões de Salvar e Compartilhar, ele simplesmente pára e não lê as opções do menu, da barra de ferramentas e o texto sendo editado.	Suporte a diferentes tipos de entrada e saída	P2, P3, P4
12	Não é possível acessar os itens dos menus usando o teclado.	Suporte a diferentes	P2

		tipos de entrada e saída	
13	Usando cores em alto contraste do sistema, ‘quando se clica em alguma função, Text Color, por exemplo, a janela que é aberta não possui nada que possa ser visualizado no monitor.	Respeito às preferências do usuário	P2
14	Os links que o usuário adiciona no seu documento ficam impossíveis de serem percebidos caso o usuário utilize cores em alto contraste do sistema.	Respeito às preferências do usuário	P2
15	Ao utilizar uma tela de resolução pequena, como 800x600, as janelas de ações (por exemplo, Inserir Tabela) não cabem totalmente na tela e, quando abertas, não podem ser fechadas. Isto pode prejudicar usuários que necessitam utilizar tais tipos de resolução – por exemplo, usuários com visão subnormal.	Suporte a diferentes tipos de entrada e saída	P2
16	Navegar usando apenas o teclado é muito ineficiente, exige que o usuário dê muitos “Tabs” para chegar a algumas opções.	Eficiência em navegação alternativa	P3
17	As teclas de atalho do navegador se confundem com as do sistema.	(não associado a qualquer heurística)	P3
18	É possível alterar o título do documento pressionando “Enter” depois que o leitor de telas o lê, mas nenhuma informação a respeito disto é dada ao usuário.	Suporte a diferentes tipos de entrada e saída	P4
19	O usuário com leitor de telas não sabe quando o leito de telas já está na caixa de edição do texto e muito menos o que está sendo digitado – não há feedback!	Suporte a diferentes tipos de entrada e saída	P4
20	Quando o usuário aumenta o tamanho da fonte no navegador usando o comando Ctrl++, os menus com muitos itens podem não aparecer por completo.	Respeito às preferências do usuário	P4
21	Se o usuário aumenta o tamanho da fonte no navegador usando o comando Ctrl++ e a mantém muito grande, ao clicar em um menu sempre o primeiro item é selecionado, ou seja, o usuário	Respeito às preferências do usuário	P2

	não consegue acessar as demais opções do menu.		
22	Não há teclas de atalho para ir ao campo de digitação do texto ou para ir ao menu.	Eficiência em navegação alternativa	P4
23	Se existe algum texto colorido, ele não será visível em alto contraste.	Respeito às preferências do usuário	P4
24	A funcionalidade de verificar ortografia não funciona caso o usuário esteja usando cores em alto contraste no sistema.	Respeito às preferências do usuário	P4

Anexo E

Resultados da avaliação de acessibilidade – segundo experimento

A tabela a seguir apresenta os problemas de acessibilidade encontrados para o Windows Live Messenger durante o segundo experimento da fase de refinamento das heurísticas, como descrito no Capítulo 4. Na coluna "Participantes", P1, P2, P3 e P4 representam os participantes que encontraram um determinado problema. Por exemplo, o primeiro problema na tabela foi identificado por todos os participantes, P1, P2, P3 e P4, ao passo que o segundo problema foi identificado apenas por um deles, P1.

Tabela E. 1. Problemas encontrados durante o segundo experimento com as heurísticas de acessibilidade.

#	**Problema**	**Heurística(s) violada(s)**	**Participantes**
1	As preferências do usuário configuradas no Windows, como cores em alto contraste e tipo de fonte, são ignoradas pelo programa desde a sua tela de login até dentro do mesmo, na lista de contatos, na janela de troca de mensagens e na janela de configuração de opções.	Respeito às preferências do usuário	P1, P2, P3, P4
2	É difícil identificar onde está o foco nos elementos do Messenger, pois o mesmo é muito discreto.	Eficiência em navegação alternativa	P1
3	A única diferença do símbolo indicando quem está online e quem está offline é a cor.	Conteúdo para todos os usuários	P4
4	A janela com atalhos às funcionalidades relacionadas a um contato só aparece quando o cursor do mouse está em cima. Até existem percursos alternativos para se acessar as mesmas funcionalidades usando apenas o teclado, mas estes percursos são muito mais longos.	Eficiência em navegação alternativa	P1

5	No histórico de mensagens trocadas, o usuário precisa clicar várias vezes nas setas para cima e para baixo do teclado, não podendo mantê-las pressionadas. Isto acaba comprometendo a eficiência de quem só pode navegar com o teclado, sem usar o mouse.	Eficiência em navegação alternativa	P3
6	O leitor de telas não funciona em todo o programa.	Suporte a diferentes tipos de entrada e saída	P1, P2, P4
7	Independente das preferências do usuário configuradas no Messenger ou no Windows, os emoticons sempre aparecem no mesmo tamanho, sempre "pequenos".	Respeito às preferências do usuário	P3
8	Usuário não consegue aumentar a fonte sozinho caso ele necessite de fontes grandes para conseguir ler as mensagens no Messenger.	Independência de uso	P3
9	Apesar de permitir outras formas de comunicação além da escrita, o conteúdo do programa é estritamente textual. Poderia haver mais símbolos ou mais figuras para facilitar seu entendimento pela população de surdos não alfabetizados.	Conteúdo para todos os usuários	P4
10	Assim que o usuário faz o *login*, uma tela é aberta com notícias do MSN. Mas o leitor de telas não a lê.	Suporte a diferentes tipos de entrada e saída	P2
11	Com o leitor de telas, é possível entrar na lista de contatos, mas o mesmo não lê o nome de quem está nela e o status dos contatos (online, offline, ocupado, etc.).	Suporte a diferentes tipos de entrada e saída	P2, P3, P4
12	Não é possível selecionar com o "Enter" do teclado um contato na lista de contatos para iniciar uma conversa.	Suporte a diferentes tipos de entrada e saída	P2
13	Quando alguém envia uma mensagem e uma janela de mensagens é aberta, o leitor de telas não lê que está nela e nem quem começou a conversa.	Suporte a diferentes tipos de entrada e saída	P2
14	O leitor de telas não lê o que o usuário escreve quando está compondo uma mensagem.	Suporte a diferentes tipos de entrada e saída	P2, P4

15	Na janela de troca de mensagens entre dois usuários, o leitor de telas não lê as mensagens recebidas.	Suporte a diferentes tipos de entrada e saída	P2, P3, P4
16	O leitor de telas não indica quando o usuário recebe uma nova mensagem.	Suporte a diferentes tipos de entrada e saída	P2
17	Quando se tenta incluir um novo contato usando o leitor de telas, uma nova janela é aberta para se inserir os dados do contato, mas o leitor de telas não lê esta janela.	Suporte a diferentes tipos de entrada e saída	P2
18	Na tela de entrada, o leitor de telas sempre lê que o campo email/login está vazio, mesmo quando o mesmo está preenchido.	Suporte a diferentes tipos de entrada e saída	P2, P3, P4
19	Na tela de entrada, o leitor de telas não lê quando se acessa o campo senha.	Suporte a diferentes tipos de entrada e saída	P2, P3, P4
20	Na tela de entrada, o leitor de telas não faz a leitura da caixa de seleção de status que o usuário deseja usar para entrar no Messenger.	Suporte a diferentes tipos de entrada e saída	P2
21	Na tela de entrada, o leitor de telas não faz a leitura das opções de declaração de privacidade, status do servidor e de armazenar a senha.	Suporte a diferentes tipos de entrada e saída	P2
22	O botão de ajuda não é lido pelo leitor de telas.	Suporte a diferentes tipos de entrada e saída	P2, P4
23	Quando a senha de acesso está errada, o Messenger abre uma janela informando, porém esta janela não é lida pelo leitor de telas.	Suporte a diferentes tipos de entrada e saída	P2, P4
24	O leitor de telas não lê a mensagem “Entrando...” que aparece assim que o usuário clica no botão de entrar.	Suporte a diferentes tipos de entrada e saída	P2

25	Ao entrar no Messenger com o status Ocupado, o leitor de telas não lê a janela que pergunta ao usuário se ele deseja alterar o status para Online.	Suporte a diferentes tipos de entrada e saída	P3
26	Caso o usuário necessite alterar a fonte na janela de troca de mensagens, precisará percorrer muitos passos usando apenas o teclado.	Eficiência em navegação alternativa	P3
27	O leitor de telas não lê o “login” e a “senha” sendo preenchidos na janela de entrada, deixando um usuário cego sem feedback.	Suporte a diferentes tipos de entrada e saída	P4
28	As figuras presentes na lista de contatos não são associadas a uma descrição textual, não sendo lidas pelo leitor de telas.	Conteúdo para todos os usuários	P4
29	É possível configurar o tamanho e o tipo de fonte usada para comunicar-se com cada um dos contatos na janela de troca de mensagens. Entretanto, se sair e entrar novamente no Messenger, estas configurações de fonte são perdidas.	Respeito às preferências do usuário	P4
30	O Messenger permite configurar o tamanho e o tipo de fonte usada para troca de mensagens, mas a mesma configuração não é aplicável para a lista de contatos, que sempre aparece com o mesmo tipo e tamanho de fonte.	Respeito às preferências do usuário	P4

Anexo F

Heurísticas de acessibilidade – versão final

Abaixo está o conjunto final de heurísticas obtidas nesta tese de doutorado:

1. **Suporte a diferentes tipos de entrada e saída**

   O sistema deve ser utilizável através dos dispositivos de entrada e saída que o usuário adotar, não importam quais sejam. Ou seja, as funcionalidades presentes no sistema devem ser acessíveis através destes dispositivos. Em especial, o sistema deve suportar a interação por meio de teclado, mouse e leitor de telas.

2. **Conteúdo para todos os usuários**

   Todo conteúdo presente no sistema deve ser compreensível para todos os usuários, ou para o maior número possível de usuários, o que pode incluir muitas vezes fornecê-lo de diferentes formas. Alguns exemplos incluem:

   Figuras: uma descrição textual da mesma (para cegos utilizadores de leitores de tela)

   Músicas e Podcasts: letra ou transcrição (para surdos que dominam a língua escrita)

   Texto escrito: língua de sinais, sinalizada na forma de animação, escrita de sinais ou substituir informação textual por ícones, animações ou outros componentes visuais (para surdos que não dominam a língua escrita)

   Vídeos e animações: língua de sinais por meio de um intérprete na tela

   Diferenciação de informação meio de cor, como campos obrigatórios de um formulário: marcação também com uso de símbolos (para pessoas com cromodeficiências).

3. **Independência de uso**

   Usuários devem ser capazes de ligar, desligar e configurar de forma autônoma todas as funcionalidades relacionadas à acessibilidade que o sistema fornece. Tais funcionalidades podem ser, por exemplo, aumentar o tamanho da fonte, habilitar o teclado virtual ou a lupa/lente de aumento, alternar para texto com alto contraste de cores, reduzir ou aumentar o tempo de varredura do teclado virtual, entre outros.

4. **Respeito às preferências do usuário**

   Usuários possuem diferentes particularidades e habilidades e, desta forma, podem necessitar de configurações específicas para as cores usadas na tela, o volume dos alto-falantes, o tamanho e o tipo de fonte dos textos, o tempo de resposta, entre outros. Portanto, o sistema deve respeitar tais configurações definidas pelo usuário, principalmente se tais configurações foram definidas de forma global (no sistema operacional ou no navegador de Internet, por exemplo).

5. **Eficiência em interação alternativa**

   O sistema deve oferecer maneiras para o usuário evitar percorrer longos caminhos ao usá-lo com tecnologias assistivas. Um sistema pode se mostrar muito ineficiente para os usuários de leitores de telas, teclados virtuais e outras tecnologias assistivas. Por exemplo, ao navegar por um site que não conhecem, muitas vezes os usuários de leitores de telas são obrigados a ler muitos parágrafos antes de chegar ao que efetivamente importa ou desejam. Assim, o sistema deve oferecer teclas de atalho para facilitar o acesso às principais funcionalidades e, no caso de conteúdos textuais, deve procurar agrupá-los de alguma forma, incluir meios de acessar rapidamente estes grupos e informar apenas o necessário por meio de um diálogo simples e direto.