



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Tuhm: Uma Ferramenta de Apoio para Testes de Usabilidade

Luan Pignata Caldas

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Edna Dias Canedo

Brasília
2021



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Tuhm: Uma Ferramenta de Apoio para Testes de Usabilidade

Luan Pignata Caldas

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof.a Dr.a Edna Dias Canedo (Orientadora)
CIC/UnB

Prof. Dr. Rodrigo Bonifácio de Almeida Prof.a Dr.a Fernanda Lima
Universidade de Brasília (UnB) Universidade de Brasília (UnB)

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 18 de outubro de 2021

Dedicatória

Dedico este trabalho primeiramente a Deus. Também dedico aos meus pais, minha família, meus amigos e todos aqueles que me apoiaram para a conclusão deste trabalho, e dedico à minha orientadora Edna Dias Canedo por todo o apoio e ajuda neste trabalho.

Agradecimentos

Agradeço a Deus pela realização deste trabalho. Agradeço também aos meus pais, minha família, minha namorada, amigos, colegas de trabalho e todos os que participaram da minha trajetória, sempre me apoiando e incentivando durante todo o curso de graduação. Agradeço aos especialistas que se dispuseram do seu tempo para a realização do estudo de caso prático, e agradeço à minha orientadora Edna Dias Canedo, por todo o apoio e auxílio prestado durante o desenvolvimento deste trabalho.

Resumo

A avaliação heurística é um processo muito importante na execução de testes de usabilidade. Através dela, o especialista pode encontrar erros que afetam a usabilidade do usuário. Entretanto, a avaliação heurística se trata de um processo moroso. Com isso, há a necessidade de se criar meios que ajudem o processo de avaliação. O objetivo deste trabalho é desenvolver uma ferramenta para apoiar o especialista de usabilidade no processo de avaliação. Foi realizada uma revisão da literatura com a finalidade de encontrar estudos relacionados às heurísticas de usabilidade e as ferramentas desenvolvidas para realizar avaliação heurística. A partir das heurísticas identificadas na literatura, foi desenvolvida uma ferramenta para auxiliar o especialista na avaliação de usabilidade. Em seguida, para validar a ferramenta foi realizado um estudo de caso onde os especialistas avaliaram a usabilidade de um aplicativo móvel. No processo de avaliação, os especialistas foram divididos em dois grupos, um grupo utilizou a ferramenta proposta durante a avaliação das heurísticas de usabilidade e o outro grupo que realizou a avaliação manual do aplicativo. A divisão em dois grupos possibilitou uma comparação dos resultados obtidos com a avaliação. Nossos achados revelam que a utilização de uma ferramenta de avaliação de usabilidade tornou o processo de avaliação mais rápido e objetivo. Além disso, possui uma fácil utilização, tendo um formato rápido de entrada de dados e uma organização e categorização dos problemas encontrados. Nossos achados também revelaram uma dificuldade dos especialistas na leitura do conjunto das heurísticas de usabilidade, pois se trata de um conjunto muito extenso, logo, foi identificada a necessidade de melhorar a apresentação das heurísticas de usabilidade. A ferramenta demonstrou-se adequada quanto ao seu objetivo, uma vez que apresentou as heurísticas de usabilidade e suas respectivas sub-heurísticas, facilitando o processo de avaliação das heurísticas de usabilidade durante a avaliação pelos especialistas. Assim, a utilização de uma ferramenta de avaliação de usabilidade apoia os especialistas de usabilidade durante o processo de avaliação heurística.

Palavras-chave: Ferramenta, Usabilidade, Avaliação Heurística, Aplicação Móvel, Heurísticas de Usabilidade

Abstract

A heuristic evaluation is a very important process in execution usability tests. Through it, the specialist can find errors that affect the user's usability. However, the heuristic evaluation is a lengthy process. Thus, there is a need to create means to help the evaluation process. The objective of this work is to develop a tool to support the usability specialist in the evaluation process. A literature review was carried out in order to find studies related to usability heuristics and the tools developed to perform heuristic evaluation. From the heuristics identified in the literature, a tool was developed to help the specialist in the usability evaluation. Then, to validate the tool, a case study was carried out where specialists evaluated the usability of a mobile application. In the evaluation process, the specialists were divided into two groups, one group used the proposed tool during the evaluation of the usability heuristics and the other group performed the manual evaluation of the application. The division into two groups allowed a comparison of the results obtained with the evaluation of the two groups. Our findings reveal that the use of a usability evaluation tool made the evaluation process faster and more objective. In addition, it is easy to use, having a fast data entry format and an organization and categorization of the problems found. Our findings also revealed a difficulty for specialists in reading the set of usability heuristics, as it is a very extensive set, therefore, the need to improve the presentation of the usability heuristics was identified. The tool proved to be adequate for its purpose, as it presented the usability heuristics and their respective sub-heuristics, facilitating the usability heuristic evaluation process during the specialist evaluation. Thus, the use of a usability evaluation tool supports usability specialists during the heuristic evaluation process.

Keywords: Tool, Usability, Heuristic Evaluation, Mobile Application, Usability Heuristics

Sumário

1	Introdução	1
1.1	Problema de Pesquisa	2
1.2	Justificativa	2
1.3	Objetivos	3
1.3.1	Objetivo Geral	3
1.3.2	Objetivo Específico	3
1.4	Resultados Esperados	3
1.5	Metodologia de Pesquisa	4
1.6	Estrutura do Trabalho Final de Curso	4
2	Embasamento Teórico	6
2.1	Usabilidade	6
2.2	Avaliação Heurística	7
2.3	Heurísticas de Usabilidade	8
2.3.1	HU1 – Visibilidade do Status do Aplicativo	9
2.3.2	HU2 – Correspondência entre o Aplicativo e o Mundo Real	11
2.3.3	HU3 – Controle do Usuário e Liberdade	13
2.3.4	HU4 – Consistência e Padrões	14
2.3.5	HU5 – Prevenção de Erros	16
2.3.6	HU6 – Minimização do <i>Cognitive Load</i> do Usuário	17
2.3.7	HU7 – Customização e Atalhos	19
2.3.8	HU8 - Eficiência de Uso e Desempenho	20
2.3.9	HU9 – Design Estético e Minimalista	22
2.3.10	HU10 – Ajuda ao Usuário para Reconhecer, Diagnosticar e Recuperar-se de Erros	23
2.3.11	HU11 – Ajuda e Documentação	24
2.3.12	HU12 – Interação Agradável e Respeitosa com o Usuário	26
2.3.13	HU13 – Privacidade	27
2.4	Trabalhos Correlatos	28

2.5	Síntese do Capítulo	30
3	Desenvolvimento	31
3.1	Requisitos do Sistema	31
3.2	Tecnologia e Arquitetura	32
3.3	Ferramenta Desenvolvida	33
3.4	Síntese do Capítulo	39
4	Estudo de Caso Prático	40
4.1	Avaliação Heurística do ESV	40
4.1.1	Procedimentos para a Avaliação Heurística	40
4.1.2	Resultados das Avaliações Heurísticas	42
4.2	Refinamentos da Ferramenta	50
4.3	Riscos à Validação e Limitações da Ferramenta	52
4.4	Síntese do Capítulo	53
5	Considerações Finais	54
5.1	Conclusão	54
5.2	Trabalhos Futuros	55
	Referências	57

Lista de Figuras

1.1	Passos da Metodologia.	4
3.1	Diagrama da arquitetura do Tuhm.	33
3.2	Tela inicial do Tuhm.	34
3.3	Tela de detalhes da avaliação, contendo a listagem dos problemas.	34
3.4	Tela de Adicionar/Editar uma Avaliação.	35
3.5	Tela de relatório da avaliação.	36
3.6	Tela de Adicionar/Editar um problema.	36
3.7	Tela de seleção de heurísticas violadas.	37
3.8	Tela de detalhes da heurística.	38
3.9	Tela de relatório dos detalhes do problema.	38
3.10	Arquivo PDF de um relatório.	39
4.1	Tempo levado para conclusão da avaliação heurística para cada especialista.	43
4.2	Número de problemas encontrados separados por heurística.	44
4.3	Número de problemas encontrados quanto a gravidade para cada especialista.	45
4.4	Tela de detalhes da avaliação após o refinamento, apresentando uma visualização da imagem do problema.	51
4.5	Tela de seleção de heurísticas violadas após o refinamento, apresentando indicador de que não houve resultados na pesquisa.	51
4.6	Tela de Adicionar/Editar um problema, com suporte para adição de imagens a partir do aplicativo Google Drive.	52

Lista de Tabelas

2.1	Comparativo entre as ferramentas existentes e a proposta neste trabalho	30
4.1	Perfil dos Especialistas da Avaliação Heurística	41
4.2	Melhorias propostas pelos especialistas para a ferramenta	50

Lista de Abreviaturas e Siglas

CNJ Conselho Nacional de Justiça.

ESV Escritório Social Virtual.

FAP-DF Fundação de Apoio à Pesquisa do Distrito Federal.

GDF Governo do Distrito Federal.

MVVM Model-View-ViewModel.

PDF Formato de Documento Portátil.

UI Interface de Usuário.

UnB Universidade de Brasília.

UX Experiência do Usuário.

Capítulo 1

Introdução

Devido a evolução das tecnologias, cada vez mais os dispositivos móveis têm se popularizado e feito parte da vida das pessoas. Isso se deve principalmente por sua capacidade de desempenhar diversas funcionalidades de forma portátil, que outrora necessitavam de instrumentos especializados. Com o grande crescimento de usuários, diversos aplicativos vêm sendo desenvolvidos. Entretanto, nem todos os aplicativos conseguem suprir as expectativas dos usuários [1]. As heurísticas de usabilidade são utilizadas como um meio de análise de aplicativos, para que se possa realizar uma avaliação e trazer um uso mais eficiente e satisfatório para o usuário [2].

O conceito de usabilidade pode ser descrito como a facilidade que os usuários possuem para aprenderem a utilizar o sistema, o quão eficientes os usuários podem ser uma vez que eles tenham aprendido a utilizar o sistema, e em como é agradável utilizar esse sistema [3]. Ou seja, o quanto o usuário pode ser eficiente e efetivo ao utilizar um aplicativo, além do seu nível de satisfação. É um requisito muito importante em termos de design, de navegação e de decisão do usuário ao se adquirir um produto [1].

Uma maneira de avaliar a usabilidade de um aplicativo é através das heurísticas de usabilidade. As heurísticas de usabilidade são um conjunto de princípios de design e uso de uma aplicação, a fim de avaliar a usabilidade. Através das heurísticas de usabilidade, especialistas em usabilidade e usuários comuns podem avaliar os aplicativos [2].

Com o passar do tempo, diversos conjuntos de heurísticas de usabilidade estão sendo propostos. Alguns com propostas mais genéricas, sem um público-alvo definido, como o trabalho proposto por Nielsen [4], enquanto outros trabalhos são mais específicos, como o trabalho de Silva et al. [5], que tem usuários idosos como público-alvo. Além disso, o analista deve se atentar ao contexto de usabilidade do dispositivo móvel. Isso ocorre porque dispositivos móveis são aparelhos que apresentam um tamanho de tela reduzido, afetando a usabilidade [6]. Há também o contexto de uso, que é o contexto em que o usuário está inserido no momento do uso, por exemplo, o usuário pode estar caminhando

enquanto utiliza o dispositivo [7].

A proposta deste trabalho é desenvolver uma ferramenta para apoiar os especialistas durante a realização dos testes de usabilidade através das heurísticas de usabilidade. Assim, será desenvolvido um aplicativo capaz de mostrar de forma clara e objetiva todo o conjunto de heurísticas de usabilidade, bem como suas sub-heurísticas. Através do aplicativo, o especialista poderá cadastrar os problemas encontrados durante a sua avaliação. Será possível também obter uma análise de gravidade e o cálculo de erros encontrados durante as avaliações, conforme proposto por Costa [8, 9].

1.1 Problema de Pesquisa

O processo de avaliação de heurísticas em um aplicativo é um processo moroso, pois são muitas heurísticas e sub-heurísticas a serem analisadas. Isso consome bastante tempo do especialista que irá realizar a avaliação, fazendo com que muitas das vezes o especialista se sintam indispostos a realizá-la. Portanto, se faz necessário criar meios que tornem o trabalho desses especialistas mais produtivo e menos cansativo.

Algumas ferramentas foram desenvolvidas, como a ferramenta Heva [10], que dá suporte a avaliação heurística para sistemas Web. A ferramenta Heva [10] teve uma grande aceitabilidade, tornando o processo mais rápido e mais satisfatório. Entretanto, a ferramenta é especializada somente na avaliação heurística de sistemas Web, não contemplando a avaliação de aplicativos para dispositivos móveis, que é o foco deste trabalho.

1.2 Justificativa

Diversas ferramentas estão sendo desenvolvidas para dar suporte à avaliação de usabilidade, como é o caso da ferramenta Heva [10] com foco em sistemas Web, mas não foi encontrada alguma com foco em aplicativos de dispositivos móveis. Além disso, há a necessidade de ser desenvolvido mais ferramentas para essas análises, tendo em vista que a falta delas torna o processo mais difícil e demorado, podendo haver uma redução no desempenho dos analistas de usabilidade [11].

Através do desenvolvimento de uma ferramenta que dê suporte aos especialistas na avaliação heurística de um aplicativo, o processo se torna mais fácil, agradável e rápido. Sendo assim, com a utilização de uma ferramenta especializada, as avaliações podem ter resultados mais satisfatórios, além de tornar o processo menos desgastante para o especialista.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma ferramenta para automatizar o conjunto de heurísticas de usabilidade para avaliação de aplicações móveis. Através dessa ferramenta, o especialista terá um suporte ferramental para apoiá-lo na execução dos testes de usabilidade em aplicativos de dispositivos móveis. Além disso, será possível também obter um relatório final, contendo os dados e estatísticas da avaliação realizada.

1.3.2 Objetivo Específico

Para atingir o objetivo geral deste trabalho, os seguintes objetivos específicos foram definidos:

- Buscar na literatura um conjunto de heurísticas de usabilidade para avaliação de aplicações em dispositivos móveis;
- Desenvolver uma ferramenta que dê suporte à avaliação de usabilidade em aplicativos de dispositivos móveis;
- Realizar um estudo de caso prático com a ferramenta, através de uma avaliação heurística de um aplicativo móvel;
- Validar a ferramenta proposta através de testes com especialistas em usabilidade.

1.4 Resultados Esperados

Através do desenvolvimento de uma ferramenta de suporte à avaliação de usabilidade em aplicações de dispositivos móveis, espera-se que a ferramenta possa ajudar o especialista a concluir a sua avaliação com mais facilidade, maior velocidade, e de forma mais agradável. Também é esperado que as avaliações possam ter maior efetividade, tendo em vista que o processo se torna menos complexo através do uso de uma ferramenta.

Os relatórios gerados pela ferramenta também tendem a ser mais detalhados, trazendo informações específicas de cada erro encontrado e estatísticas gerais da avaliação. As estatísticas gerais também tendem a ter uma confiabilidade maior em seus dados, tendo em vista que todo o cálculo será automatizado pela ferramenta.

1.5 Metodologia de Pesquisa

Para a execução deste trabalho, foi utilizada uma metodologia baseada em evidências que utiliza estudos primários e secundários. Através dos estudos primários, é possível conduzir estudos experimentais sobre a ferramenta a ser desenvolvida. Já os estudos secundários têm o papel de realizar uma revisão literária sobre o problema de pesquisa, obtendo resultados mais precisos e confiáveis, por meio de trabalhos já realizados anteriormente.

Foi realizada uma Revisão de Literatura como um estudo secundário, com o intuito de classificar as soluções propostas para o problema de pesquisa. Através da Revisão de Literatura é possível localizar, analisar, sintetizar e interpretar as pesquisas disponíveis na área de interesse [12]. Através desse estudo, foi possível agregar um conhecimento do problema e suas soluções, para que fosse possível determinar uma solução mais robusta para o foco deste trabalho.

Em seguida, a solução foi avaliada por intermédio de estudos experimentais (estudos primários). Através desses estudos, a solução foi testada por especialistas, e foram coletadas informações de uso. Com essas informações, foi realizada uma análise entre especialistas que utilizaram a solução e os que não utilizaram, com o intuito de se obter uma certificação de que os resultados foram alcançados.



Figura 1.1: Passos da Metodologia.

1.6 Estrutura do Trabalho Final de Curso

Este trabalho está organizado da seguinte maneira:

- No Capítulo 2 é apresentado o embasamento teórico do trabalho. São apresentados os principais conceitos, e o conjunto de heurísticas que será utilizado no trabalho;
- No Capítulo 3 são apresentados os requisitos para o desenvolvimento da ferramenta. Também é mostrada a arquitetura e as tecnologias que foram utilizadas para o desenvolvimento da ferramenta, e há uma apresentação da ferramenta.

- No Capítulo 4 é realizado um estudo de caso prático com a ferramenta, sendo apresentados os resultados e as limitações da ferramenta.
- No Capítulo 5 são apresentadas a conclusão deste trabalho e os trabalhos futuros.

Capítulo 2

Embasamento Teórico

Neste capítulo são apresentados os conceitos e definições de Usabilidade, Avaliação Heurística e o Conjunto de Heurísticas de Usabilidade que será utilizado neste trabalho. Além disso, este Capítulo apresenta alguns trabalhos relacionados a esta pesquisa.

2.1 Usabilidade

Nielsen [4] mostra a usabilidade como um dos elementos para se alcançar a aceitabilidade de um sistema. Com isso, é possível afirmar que tal elemento possibilita saber se um sistema é capaz de satisfazer todas as necessidades e exigências dos usuários. Nielsen também definiu cinco atributos como uma forma de definir a usabilidade, e que podem ser utilizados no auxílio da avaliação de um sistema. São eles:

1. *Facilidade no Aprendizado*: É a facilidade no aprendizado da utilização do sistema, diz respeito ao tempo e esforço gasto pelo usuário para que ele possa aprender a utilizar o sistema, fazendo com que o usuário se sinta mais habituado ao utilizar o sistema;
2. *Eficiência do Usuário*: A eficiência do usuário leva em consideração o tempo necessário para que o usuário possa realizar uma tarefa no sistema, trazendo uma maior produtividade do usuário;
3. *Facilidade na Memorização*: A facilidade na memorização leva em consideração a dificuldade que o usuário tem para repetir uma determinada ação, uma vez que ele já a tenha aprendido;
4. *Proteção Contra Erros*: A proteção contra erros é o atributo no qual se refere a organização dos elementos de um sistema com a finalidade de evitar com que erros sejam cometidos pelos usuários;

5. *Satisfação do Usuário*: É a satisfação pela utilização do sistema, é o quão satisfeito o usuário pode se sentir ao utilizar um determinado sistema.

Além disso, há os fatores de usabilidade, os quais estão diretamente relacionados a usabilidade de um sistema, podendo interferir em sua programação visual, assim como trazer danos à relação usuário-sistema. Harisson et al. [13] define três fatores de usabilidade, sendo estes:

1. *Usuário*: O fator em questão carrega consigo duas condições essenciais para o desenvolvimento de um sistema, sendo estas, condições físicas e condições voltadas para a experiência de uso do sistema. Assim, as condições físicas, quando estas apresentam alguma limitação ao usuário, vão direcionar a forma com que o usuário utiliza o sistema. Já as condições voltadas para as experiências de uso do sistema, vão demonstrar a preferência de um tipo específico de interface, onde os usuários habituados ao sistema irão preferir uma configuração gráfica mais direta, e aqueles sem experiência anterior vão preferir uma configuração gráfica mais instrutiva;
2. *Tarefa*: O usuário ao utilizar determinado sistema, apresenta um objetivo a ser realizado. Portanto, um excesso de funcionalidades pode aumentar a complexidade do sistema, fazendo com que o usuário esteja mais suscetível a erros;
3. *Contexto de Uso*: Este fator de usabilidade, tem como característica a relação do usuário com o ambiente na qual a aplicação será utilizada, assim como a relação do usuário com os objetos e pessoas ao seu redor. No estudo de Bergman e Vainio [7], é demonstrado como essa relação entre usuário e ambiente interfere diretamente na utilização de aplicações móveis, onde foi identificado uma diminuição da velocidade média da caminhada exercida pelo usuário enquanto este, simultaneamente, usava uma aplicação móvel.

2.2 Avaliação Heurística

Heurísticas são diretrizes de usabilidade que descrevem princípios desejáveis da interação e da interface. Elas são aplicadas em interfaces de um sistema, sendo conduzidas por especialistas em testes de usabilidade. Esses especialistas analisam a interação e a interface do sistema, sendo guiados pelas heurísticas de usabilidade [14]. Este processo é conhecido como *Avaliação Heurística*, e é um dos métodos de avaliação de usabilidade mais utilizados [15].

O procedimento de avaliação é realizado por meio de um grupo de especialistas. Eles analisam a interface do usuário utilizando as heurísticas de usabilidade, que são as normas

de usabilidade. Os especialistas analisam as interfaces em busca de problemas, e ao encontrá-los, são informados junto às heurísticas que foram infringidas [16], podendo ser classificado com um grau de gravidade. Os seguintes graus de gravidade são propostos por Nielsen [4]:

0. *Inexistente*: Não concordo que seja um problema de usabilidade;
1. *Problema Cosmético*: Não precisa ser corrigido, a menos que haja tempo extra disponível no projeto;
2. *Problema Pequeno*: O conserto deste problema pode receber baixa prioridade;
3. *Problema Grande*: Importante para corrigir, por isso deve ser dada alta prioridade;
4. *Problema Catastrófico*: É de extrema importância consertá-lo antes de se lançar o produto. Se mantido, o problema provavelmente impedirá o usuário de realizar suas tarefas e alcançar seus objetivos.

O mais recomendado é que cada especialista faça a sua análise de forma independente [14]. Assim, quando há vários especialistas, todos devem fazer suas análises de forma individual. Isso traz uma imparcialidade e uma precisão maior, sendo os resultados compartilhados somente ao final das análises. Durante a avaliação, os especialistas devem analisar o sistema repetidas vezes, conforme os parâmetros definidos para ele. Por isso, o contexto de uso se faz necessário, pois leva em consideração as particularidades do sistema. Assim, o especialista é capaz de produzir resultados mais precisos.

Ao final das avaliações, os resultados obtidos pelos especialistas devem ser reunidos e um relatório deve ser gerado contendo uma visão geral dos problemas identificados. Logo após, os especialistas podem discutir sobre os problemas e podem realizar a modificação do grau de gravidade de cada problema. Em seguida, são decididas as prioridades dos problemas a serem resolvidos com base nos resultados alcançados no relatório final.

2.3 Heurísticas de Usabilidade

No trabalho de Costa et al. [8, 9] foi realizada uma Revisão Sistemática de Literatura (RSL) para identificar na literatura os principais trabalhos sobre Heurísticas de Usabilidade. Os autores identificaram 12 estudos primários. A partir da análise desses estudos, Costa et al. [8, 9] propuseram 13 heurísticas de usabilidade, e os autores realizaram duas avaliações. Através das avaliações, foram realizados refinamentos nas heurísticas propostas, resultando em um conjunto de 13 heurísticas e 183 sub-heurísticas de usabilidade. As principais diferenças para as heurísticas de Nielsen [4] estão no fato de algumas sub-heurísticas serem específicas para dispositivos móveis, além de levar em consideração o

contexto de uso, a privacidade e o *cognitive load* do usuário. As heurísticas resultantes são apresentadas nas próximas subseções.

2.3.1 HU1 – Visibilidade do Status do Aplicativo

Definição: O aplicativo deve manter o usuário informado sobre todos os processos, interações do usuário com o sistema e mudanças de estado dentro de um período de tempo razoável.

Explicação: Por meio da interação com o aplicativo, o usuário deve ser capaz de executar diferentes tarefas. Se as ações do usuário levarem a uma mudança de estado do aplicativo (status atual (onde estou), possíveis mudanças (o que eu posso fazer aqui) e/ou *feedback* (o que acabou de acontecer)), o usuário deve ser informado de alguma forma (sons, mensagens e/ou animações na tela). As ações do usuário, como o de gestos (toque na tela e/ou arraste da tela), devem dar um *feedback* de forma visível, clara e concisa.

Benefícios: O usuário pode ter uma melhor experiência de uso quando o aplicativo informar sobre mudanças de estado, de maneira precisa e apropriada. Outro benefício está relacionado à conscientização do usuário ao utilizar o aplicativo, dado que ele saberá se teve ou não alguma alteração significativa do estado da aplicação.

Problemas: É importante distinguir, ao aplicar esta heurística, que o usuário pode ter problemas com a ausência de *feedback* devido à falta de implementação desta funcionalidade ou problemas de desempenho do aplicativo causados por um grande número de processos simultâneos e/ou devido a bateria estar com pouca carga (ativando o modo econômico de energia), afetando o desempenho do aplicativo e podendo ocasionar a falta de *feedback*.

Sub-heurísticas:

1. Cada tela da aplicação começa com um título ou um cabeçalho que descreve o conteúdo da tela?
2. Um único ícone selecionado é claramente visível quando está cercado por ícones não selecionados?
3. Se o sistema receber informações importantes de ações de fundo (por exemplo, notificações internas do aplicativo, processo de envio *upload* ou *download*, entre outros), o sistema responde (por exemplo, vibrar, soar) alertando os usuários?
4. Se os usuários precisarem navegar entre várias telas, elas possuem títulos significativos?
5. Se notificações *pop-up* são usadas para exibir mensagens de erro, elas permitem que o usuário veja o campo com o erro?

6. Existe alguma forma de *feedback* do sistema para cada ação do usuário?
7. Se o usuário está rolando para o limite de um elemento (por exemplo, *listview*), há alguma sugestão visual?
8. Se o sistema contém *splash screens* (primeira tela mostrada durante a execução de um aplicativo), há algum *feedback* visual do que o sistema está fazendo enquanto as telas iniciais são exibidas?
9. Se os modos sobrescrever e inserir estiverem disponíveis, há uma indicação visual de qual deles o usuário está?
10. O sistema fornece divulgação de progresso informativo ao executar uma ação que o usuário precisa aguardar (porcentagem de conclusão ou tempo de espera para completar a tarefa)?
11. Se forem utilizados menus expansíveis, os rótulos dos menus indicam que eles se expandem para um conjunto de opções?
12. Depois que o usuário conclui uma ação (ou grupo de ações), o aplicativo indica se é possível iniciar uma nova ação (ou grupo de ações)?
13. Existe *feedback* visual na tela sobre quais opções são selecionáveis (permitem interação do usuário)?
14. Se vários objetos na tela puderem ser selecionados, existe um *feedback* visual sobre quais objetos já foram selecionados?
15. Existe *feedback* visual quando os objetos são movidos de um ponto da tela para outro?
16. Existe *feedback* visual quando o usuário realiza a ação de atualizar (*refresh*) a tela?
17. O usuário pode identificar o estado do sistema, e as alternativas de ação disponíveis, simplesmente ao olhar para as informações exibidas na tela da aplicação?
18. As barras de status do sistema operacional são, na maioria das vezes (ou sempre) visíveis, exceto para conteúdo multimídia?
19. Os botões do sistema operacional (por exemplo, botão voltar e botão inicial) são a maioria (ou sempre) visíveis, exceto para conteúdo multimídia?
20. Se houver atrasos observáveis (mais de cinco segundos) no tempo de resposta do sistema, o usuário é informado sobre o estado da aplicação?
21. Os tempos de resposta são apropriados para a tarefa?

- Digitação, *swipe*, *scrolling*: 1-50 milissegundos;
- Tarefas simples e frequentes: menos de 1 segundo;
- Tarefas comuns: 2-4 segundos;
- Tarefas complexas: 8-12 segundos.

22. Os tempos de resposta são adequados ao processamento cognitivo do usuário?

- A continuidade de pensamento é necessária e a informação deve ser lembrada em várias respostas: menos de dois segundos;
- Altos níveis de concentração não são necessários e não é necessário lembrar informações (minimizar a carga de memória do usuário): de dois a quinze segundos.

2.3.2 HU2 – Correspondência entre o Aplicativo e o Mundo Real

Definição: O aplicativo deve falar o idioma dos usuários e não em termos técnicos de sistema. O aplicativo deve exibir as informações de forma familiar ao usuário (de acordo com o contexto externo em que a aplicação está inserida), seguindo uma ordem lógica e natural.

Explicação: Atualmente, os aplicativos têm vários modos de interação, em que os usuários podem executar tarefas de maneiras intuitivas, imitando regras de interação do mundo real. Por exemplo, ao rolar para baixo uma lista longa, se o usuário “deslizar” com certa velocidade, a lista continuará a se mover, imitando o efeito da inércia. Outro exemplo é com relação ao gesto multitoque em um aplicativo, ao executar um gesto de deslizar para a esquerda o sistema deve trazer o próximo objeto do lado direito, ou seja, mostrando o próximo objeto do lado direito (exemplo: funcionalidade de carrossel). Espera-se que cada interação mostre uma resposta semelhante à esperada no mundo real. Além disso, a linguagem (texto ou ícones) deve estar relacionada ao mundo real e/ou a termos familiares ao usuário, utilizando como base o contexto externo à que a aplicação está inserida, por exemplo: aplicações de controle médico, os termos utilizados por essa aplicação devem corresponder ao domínio/contexto médico ao qual está inserido.

Benefícios: O usuário, ao reconhecer conceitos do mundo real no aplicativo, terá uma barreira menor de adaptação ao utilizar o sistema e irá interpretar corretamente as informações fornecidas na aplicação e dessa forma obterá uma maior eficiência em suas ações, pois serão apresentadas em uma ordem lógica e natural. Outro benefício é a minimização de erros cometidos pelo usuário durante o uso do sistema garantindo também

uma maior eficiência de uso, pois o usuário estará mais familiarizado com o aplicativo e erros de interação do usuário na aplicação serão minimizados.

Problemas: Ao aplicar esta heurística, não deve ser confundido com “HU4 – Consistência e Padrões”. Se algumas partes do sistema estão em um idioma diferente, não é uma questão relacionada a esta heurística, mas uma questão de consistência interna da aplicação (heurística HU4), enquanto a consistência externa da aplicação (entre aplicativo e ambiente ao qual ela está inserida) está relacionada a heurística HU2.

Sub-heurísticas:

23. As metáforas (por exemplo, ícones que correspondem a ações) são utilizadas?
24. Os ícones são familiares ao usuário, dado o contexto em que a aplicação está inserida?
25. Os elementos da tela são ordenados da maneira lógica, de forma a considerar as características do usuário e o contexto de uso da aplicação?
26. As teclas de função estão nomeadas (*labeled*) de forma clara e distinta?
27. Os elementos da interface, relacionados e interdependentes, aparecem na mesma tela?
28. Nas telas de entrada de dados, as tarefas são descritas na terminologia familiar aos usuários (considerando o contexto que a aplicação está inserida)?
29. Se formas são utilizadas como uma dica visual (por exemplo, uma forma de “bola de basquete”), ela combina com convenções culturais (é exibida de forma a lembrar uma “bola de basquete”)?
30. As terminologias dos textos utilizados na aplicação são consistentes com o domínio das tarefas do usuário (são significativas ao usuário)?
 - Use a nomenclatura no domínio específico;
 - Empregue o jargão do usuário e evite o jargão do sistema.
31. Elementos interativos da aplicação possuem correspondência com o mundo real (por exemplo, física de rolagem de lista)?
32. O sistema alinha automaticamente o formato para valores numéricos (por exemplo, inserindo espaços à direita, entrelinhas, vírgulas e/ou um símbolo monetário)?

2.3.3 HU3 – Controle do Usuário e Liberdade

Definição: O aplicativo deve permitir que o usuário desfça e refaça suas ações para uma navegação clara e deve fornecer ao usuário uma opção de saída de estados indesejáveis do sistema.

Explicação: Quando o usuário cometer um erro ao inserir um texto, modificando opções de configuração ou apenas atingindo um estado indesejado, o sistema deve fornecer “saídas de emergência” apropriadas. Essas saídas devem facilmente permitir que o usuário passe de um estado indesejado para um desejado. Além disso, o aplicativo deve permitir que o usuário desfça e refaça suas ações de uma maneira simples e intuitiva, já que os usuários podem executar gestos indesejados na tela sensível ao toque de um *smartphone* (por exemplo: toque, *swipe*, etc.), o que pode levar o aplicativo a um outro estado.

Benefícios: Por ter controle sobre o sistema, o usuário tem uma maior sensação de liberdade, conseqüentemente uma melhor experiência de uso e uma maior eficiência no uso do aplicativo.

Problemas: Esta heurística não deve ser confundida com o conceito de flexibilidade e eficiência de uso. Embora alguns efeitos de controle estejam relacionados a uma melhor eficiência (como apontados nos Benefícios acima), eles são conceitos diferentes. As heurísticas “HU5 – Prevenção de Erros” e “HU10 – Ajuda ao Usuário para Reconhecer, Diagnosticar e Recuperar-se de Erros” são relacionadas a mostrar, respectivamente, formas de prevenir erros e de informar o usuário sobre o erro e também sobre como recuperar-se do erro, dessa forma, a heurística “HU3 – Controle do Usuário e Liberdade” destina-se a reparar ou corrigir erros, dando ao usuário a chance de desfazer ou refazer suas ações e ter controle sobre os recursos do sistema.

Sub-heurísticas:

33. O usuário pode interagir com o sistema continuamente (sem enfrentar travamentos ou congelamentos do sistema)?
34. Quando a tarefa de um usuário é concluída, o sistema aguarda um sinal do usuário antes de finalizá-la de fato?
35. O teclado virtual é exibido apenas quando necessário?
36. Caso o teclado virtual seja exibido, o usuário é capaz de minimizá-lo?
37. Os dados pessoais (dados sensíveis) do usuário podem ser alterados?
38. Dados gerais de cadastro (caso o aplicativo possua essa funcionalidade) do usuário podem ser alterados?

39. Se uma caixa de diálogo estiver aparecendo, a localização do botão positivo (por exemplo, botão “OK”, botão “Próximo”) no lado direito e do negativo (por exemplo, botão “Cancelar”, botão “Voltar”) esta à esquerda? (Colocar botões positivos à direita dá uma sensação de continuar e progredir na tarefa, ao passo que colocar os botões negativos à esquerda dão uma sensação de reverter a tarefa. Isso também é mais fácil de compreender de relance);
40. Se uma caixa de diálogo estiver sendo exibida, o usuário pode ser dispensado tocando em alguma área fora da caixa de diálogo?
41. As notificações indicam que o usuário está no controle (dando opções de ações)?
42. Existe uma função “desfazer” a nível de uma única ação, uma entrada de dados e um grupo completo de ações?
43. Os usuários podem cancelar operações em andamento?
44. Os usuários podem facilmente reverter suas ações?
45. Se os usuários puderem voltar a tela anterior, poderão alterar sua opção ou algum dado?
46. Se *listviews* forem longas (por exemplo, mais de sete itens), os usuários podem selecionar quaisquer itens simplesmente ao rolar a lista?
47. As teclas de função que podem causar sérias consequências ao usuário, têm o recurso de desfazer a ação?
48. O sistema usa transições para mostrar relações entre telas?
49. Os usuários podem digitar antecipadamente em campos de entrada de dados, caso não tenha preenchido os campos anteriores?
50. As alterações de caracteres são permitidas nos campos de entrada de dados?
51. Usuários podem avançar e retroceder entre campos ou opções de entrada de dados?
52. Caso o aplicativo possibilite o uso de um ponteiro de mouse, os usuários têm a opção de clicar em itens da interface?

2.3.4 HU4 – Consistência e Padrões

Definição: O aplicativo deve seguir as convenções estabelecidas, permitindo o usuário realizar as suas tarefas de maneira familiar, padronizada e consistente.

Explicação: Muitas vezes, diferentes partes do aplicativo que estão relacionadas e que devem ser semelhantes têm design ou lógica de utilização diferentes. Em geral,

todo conceito apresentado de maneira contrastante à concepção do conceito pelo usuário produz confusão em algum grau. Essa confusão pode levar a uma diminuição da eficiência de uso ou uma baixa satisfação, entre outros efeitos colaterais. Um gesto específico de interação do usuário na interface do sistema deve ter o mesmo comportamento em todo o aplicativo, fazendo com que o modelo conceitual do usuário da possível ação/interação com a aplicação seja consistente a convenções estabelecidas internamente e externas a aplicação (por exemplo, seguir padrões de controle de gestos já bem estabelecidos em aplicativos móveis de jogos eletrônicos de luta). Além disso, é melhor evitar conflitos com as operações acionadas por um gesto no aplicativo e as operações relacionadas aos gestos padrão do sistema operacional. Dadas todas essas questões, espera-se que o sistema siga padrões e convenções, estabelecidos internamente pela aplicação ou externos a aplicação, para obter uma interface intuitiva e amigável.

Benefícios: O aplicativo que segue convenções e padrões faz com que novos usuários tenham uma curva de aprendizado mais baixa enquanto utilização do sistema, que por exemplo se beneficia da experiência obtida com o uso de outros aplicativos. Assim, erros cometidos pelo usuário nas interações com o aplicativo serão minimizados, pois o usuário estará familiarizado com a navegação do sistema.

Problemas: Quando se fala em consistência, um aspecto clássico é a consistência da linguagem. Às vezes há palavras que não têm uma tradução adequada, principalmente quando relacionadas à tecnologia. Essas palavras podem ser difíceis de traduzir ou até perder o significado. Portanto, manter algumas palavras em outro idioma pode não ser um problema de “consistência”, mas manter um parágrafo inteiro é definitivamente um.

Sub-heurísticas:

53. Padrões de formatação da indústria ou da empresa são seguidos de forma consistente em todas as telas do aplicativo?
54. Quando notificações implicam uma ação necessária, o texto da mensagem é consistente com essa ação?
55. Palavras caso estejam abreviadas, seguem o mesmo padrão de abreviação?
56. Os padrões de design da plataforma (por exemplo, as diretrizes de design do iOS e Android) são seguidos?
57. Cada tela possui um título?
58. Todas as telas do sistema são exibidas de forma consistente com todos os dispositivos de diferentes plataformas/fabricantes (por exemplo, iPhone, Samsung Galaxy)?
59. Existe uma localização consistente do menu, na interface do aplicativo, em todo o sistema?

60. Existe consistência de tipografia em todo o aplicativo?
61. Existe um design consistente em elementos de entrada (por exemplo, caixa de texto, menu suspenso, etc.)?
62. Existe um design consistente no tamanho físico dos elementos da tela (tamanho da fonte, tamanho do elemento em relação a outros)?
63. A aparência da fonte do sistema (por exemplo, tamanho, tipo de letra) pode ser alterado para ser consistente com a aparência da fonte do sistema operacional (por exemplo, iOS ou Android)?
64. As cores são consistentes em todo o aplicativo?
65. A estrutura de um valor, na entrada de dados do aplicativo, é consistente em todo o sistema?
66. O sistema de navegação do aplicativo é padronizado?
67. A interação por gestos em elementos semelhantes da interface é padronizada em todo o aplicativo?
68. A estrutura de um valor na entrada de dados é consistente em todo o aplicativo?
69. Caso o sistema contenha várias telas para entrada de dados, todas elas têm o mesmo título?

2.3.5 HU5 – Prevenção de Erros

Definição: Elimine as condições propensas a erros e apresente ao usuário uma opção de confirmação com informações adicionais antes dele se comprometer com a ação.

Explicação: O aplicativo deve ser explícito em relação a cada opção e funcionalidade disponível ao usuário. Considerando um espaço de tela limitado, isso pode ser um grande desafio. Assim, os ícones desempenham um papel muito importante, como dar um retorno ao usuário quando ele interage, e mesmo que a parte visível do ícone possa ser pequena, deve haver algum espaço extra de destino invisível que, se um usuário atingir esse espaço, a interação ainda ocorrerá. Infelizmente, às vezes uma pequena imagem não é suficiente para descrever em detalhes uma função ou algo similar, e para corrigir isso, o sistema deve fornecer informações adicionais sobre a demanda do usuário. A informação deve ser claramente exibida, tentando evitar longas sequências de diálogo. Além disso, o usuário deve ser avisado, especialmente quando ocorre algumas ações que podem ter efeitos indesejados ao usuário.

Benefícios: Ao evitar que o usuário cometa erros, o mesmo gastará menos tempo tentando resolvê-los, o que pode levar a uma maior eficiência.

Problemas: Este conceito não deve ser confundido com “HU10 – Ajuda ao Usuário para Reconhecer, Diagnosticar e Recuperar-se de Erros”. Se o usuário puder acionar um erro, isso é um problema de prevenção.

Sub-heurísticas:

70. O sistema impede que usuários cometam erros sempre que possível?
71. Usuários são solicitados a confirmar comandos que tenham consequências drásticas e destrutivas?
72. É evitado o uso de textos muito longos, que não são mostrados completamente (por exemplo, utilizam no final “...”), na aplicação? (informações importantes podem estar ocultas e levar o usuário a cometer erros).
73. Quando o usuário opta por ficar logado, ele recebe uma mensagem informando sobre os possíveis riscos (caso exista)?
74. Objetos na tela têm o tamanho fácil de tocar?
75. Objetos tocáveis (por exemplo, botões) da tela estão colocados muito próximos?
76. Existem diferenças visuais entre objetos de interação (por exemplo, botões) e objetos de informação (por exemplo, rótulos, imagens)?
77. Embora a parte visível do elemento da tela possa ser pequena, há algum espaço de destino invisível que caso o usuário o atinja, a ação ainda será realizada?
78. Opções de menu são lógicas, distintas e mutuamente exclusivas?
79. Teclas de função que podem causar consequências mais graves estão em posições difíceis de serem alcançadas?
80. Na interface de entrada de dados, o aplicativo informa em tempo real o usuário sobre erros, caso insira a informação no padrão errado?
81. Os tipos de entrada de dados são apropriados para tipos de informação (por exemplo, usar o tipo de entrada numérico para informações numéricas)?
82. Campos de entrada de dados da tela contêm valores padrão quando apropriado?

2.3.6 HU6 – Minimização do *Cognitive Load* do Usuário

Definição: O aplicativo deve oferecer objetos visíveis, ações e opções para evitar que os usuários tenham que memorizar informações de uma interface para a outra.

Explicação: A memória humana de curto prazo é limitada, portanto o usuário não deve ser forçado a lembrar informações de uma parte do sistema para outra. Instruções sobre como usar o sistema devem ser visíveis ou fáceis de obter. Ao usar aplicativos executados em dispositivos móveis, os usuários geralmente estão executando outras tarefas ao mesmo tempo em que estão usando o aplicativo, como aplicativos de monitoramento de exercícios usados durante a atividade física, a interface deve ser adaptada a esse contexto de uso para que a carga cognitiva (*Cognitive Load*) do usuário seja minimizada durante o uso do sistema.

Benefícios: Ao reduzir a carga de memória (*Cognitive Load*), a capacidade mental e o esforço do usuário também são reduzidos e o usuário pode se concentrar em executar uma única tarefa com mais eficiência.

Problemas: Para esta heurística específica, o avaliador pode encontrar alguns problemas relacionados à “HU5 – Prevenção de Erros”. A questão principal aqui é observar que essa heurística está fortemente relacionada à sobrecarga de informações. Não se trata da disponibilidade de informações, mas da quantidade de informações que o usuário precisa memorizar para usar o sistema corretamente e com mais eficiência, com base no contexto de uso da aplicação.

Sub-heurísticas:

83. Todos os dados que o usuário precisa são exibidos em cada etapa de uma sequência de transações (várias telas)?
84. Os elementos da tela foram agrupados em zonas lógicas e cabeçalhos foram usados para distinguir as zonas?
85. A primeira palavra (quando disponível) de cada escolha (elemento interativo da interface) do usuário é a mais importante?
86. O sistema fornece mapeamento, ou seja, os relacionamentos entre gestos e ações são intuitivos?
87. Nos menus de navegação, o número de itens e termos (por item) são controlados para evitar a sobrecarga da memória cognitiva?
88. Para contextos de uso, em que o usuário necessite destinar sua atenção a uma outra tarefa em paralelo, a interface é adaptada para reduzir a carga de memória do usuário?
89. Os campos de entrada de dados obrigatórios estão claramente marcados?
90. O sistema fornece uma entrada de exemplo para informações complexas ou específicas do formato?

91. As solicitações, sugestões e mensagens são colocadas em locais que o usuário provavelmente estará olhando na tela (locais intuitivos para mostrar determinada informação)?
92. Áreas de texto na tela têm “espaço para respirar” ao redor delas?
93. Existe uma distinção visual óbvia entre o “escolha um” e “escolha muitos” em uma lista de elementos?
94. Entre a descrição de um campo de entrada de dados e o campo em si há um espaço mínimo que os separaram?
95. Os campos opcionais de entrada de dados estão claramente destacados?
96. O realce de cor é usado para chamar a atenção do usuário, quando necessário?
97. Objetos interativos da interface, inativos, estão esmaecidos ou com coloração mais fraca em relação as outras cores?
98. Tamanho, fonte em negrito ou sublinhado, cor, sombreamento ou tipografia, são utilizados para mostrar hierarquia entre os itens da tela?
99. As bordas são usadas para identificar grupos de objetos ou informações significativos?
100. A mesma cor foi usada para agrupar elementos relacionados?
101. A cor é usada em conjunto com alguma outra sugestão redundante?
102. Há bom contraste de cores e brilho entre as cores da imagem e do plano de fundo?
103. As cores claras, brilhantes e saturadas foram usadas para enfatizar os dados importantes em relação as cores mais escuras?
104. As telas de entrada de dados indicam quando os campos são opcionais?

2.3.7 HU7 – Customização e Atalhos

Definição: O aplicativo deve fornecer opções de configurações básicas e avançadas, de modo a permitir a criação e personalização de atalhos para ações frequentes.

Explicação: Cada usuário tem suas próprias necessidades e tentar satisfazer todas elas com um menu ou interface padrão pode ser um desafio. Assim, considere permitir que os usuários criem seus próprios atalhos e personalizações para boa parte do sistema. Através do acesso a opções de configuração avançadas, os usuários experientes podem melhorar sua eficiência de uso e novos usuários podem ter uma sensação de propriedade

mais profunda. É importante ter em vista as características do público-alvo do aplicativo, pois esse mapeamento permite implementar opções de personalizações mais ou menos avançadas, por exemplo.

Benefícios: O usuário pode personalizar algumas ações e/ou recursos com um profundo senso de propriedade. O mesmo é verdadeiro, facilitando o acesso do usuário a recursos que são mais usados e melhor adaptando o aplicativo ao perfil do usuário. Os atalhos contribuem para atingir com eficiência as metas e, assim, fornecer benefícios reais para o usuário, como maximizar a eficiência da bateria.

Problemas: Claramente, existe um limite em termos de personalização. Modificações estéticas podem ser possíveis na maioria das partes do sistema, mas algumas coisas não são alcançáveis. Mudanças significativas devem ser analisadas caso a caso quanto à validade de sua implementação, e é necessário considerar o perfil do público-alvo na análise. Não confundir esta heurística com “HU8 – Eficiência de Uso e Desempenho”, a HU7 se refere a customização de partes da aplicação para atender usuários mais avançados e/ou iniciantes. A HU8 se refere a performance geral da aplicação e da minimização de passos para uso de determinada funcionalidade.

Sub-heurísticas:

105. Os usuários são os iniciadores de ações, e não os que respondem?
106. Nas configurações do aplicativo é possível customizar alguma parte do sistema?
107. Se o sistema oferece suporte a usuários iniciantes e especialistas, vários níveis de detalhes de mensagens de erro estão disponíveis?
108. Se o sistema oferece suporte a usuários iniciantes e especialistas, as customizações são adequadas com as características do público-alvo (por exemplo, customizações significativas para usuários que utilizam o sistema em um contexto de uso médico)?
109. Se o sistema oferece suporte a usuários iniciantes e especialistas, elementos da interface podem ser customizáveis?
110. Se o sistema oferece suporte a usuários iniciantes e especialistas, elementos de navegação podem ser customizáveis?

2.3.8 HU8 - Eficiência de Uso e Desempenho

Definição: O dispositivo deve ser capaz de carregar e exibir a informação em um período de tempo razoável e minimizar as etapas necessárias para executar uma tarefa (número de etapas a serem tomadas pelo usuário para atingir uma meta). Animações e transições devem ser exibidas sem problemas e de forma fluida.

Explicação: A combinação entre recursos de hardware e necessidades de software nem sempre é a melhor. Espera-se que o aplicativo básico seja compatível com o hardware, especialmente com recursos de processamento, para evitar telas pretas e longos períodos de tempo de espera. Além disso, as animações, efeitos e transições devem ser exibidos de forma fluida e sem interrupções. Outro ponto crítico é o comprimento da sequência de etapas para executar uma tarefa. Tarefas complexas, potencialmente perigosas ou não frequentes podem conter várias etapas como reforço de segurança. Tarefas simples, frequentes ou básicas do sistema (como as principais funcionalidades do aplicativo) devem ser curtas, ou seja, ter poucas etapas para acessar e concluir.

Benefícios: Menores tempos de resposta e melhor desempenho, o que leva a um sistema eficiente.

Problemas: Os problemas relacionados ao desempenho do hardware devem ser separados dos problemas relacionados ao desempenho da rede. Mesmo que esses problemas afetem a usabilidade, eles não fazem parte do escopo desta pesquisa, considerando que eles são afetados por diversos fatores complexos. Em relação à duração de uma sequência de etapas para executar uma tarefa, o limite entre normal e excessivo é um assunto subjetivo. O avaliador deve usar seus próprios critérios.

Sub-heurísticas:

111. As telas iniciais que não fazem nada (nenhuma tarefa em segundo plano, apenas mostre a imagem ou o vídeo) foram evitadas?
112. Os menus utilizados com maior frequência estão nas posições mais acessíveis?
113. O sistema utiliza informações do dispositivo como data e hora e geolocalização como dados de entrada quando necessário?
114. Em um formulário de entrada de dados, o usuário pode mover o foco de uma caixa de texto para outra caixa de texto pressionando “próximo” no teclado virtual?
115. O sistema permite que os usuários interajam com elementos da interface deslizando, gesticulando ou “beliscando”, em vez de apenas tocar (por exemplo, usuários podem apertar o elemento de imagem para aumentar e diminuir o zoom, usuários podem deslizar para a esquerda para ir a página anterior)?
116. Se a lista for muito longa, o sistema fornece ferramentas para filtrar itens ou rolar mais rapidamente?
117. Animações de transição entre telas e/ou de elementos da interface são fluidos?
118. As principais funcionalidades da aplicação são facilmente acessadas (poucos passos a serem realizados)?

119. Em campos de pesquisa é permitido acessar o histórico de forma a permitir que usuários selecionem informações digitadas anteriormente?

2.3.9 HU9 – Design Estético e Minimalista

Definição: O aplicativo deve evitar exibir informações indesejadas que sobrecarreguem a tela.

Explicação: Os textos do aplicativo não devem conter informações irrelevantes ou raramente necessárias. Cada unidade extra de informação em uma interface compete com outras unidades de informação relevantes que possam ter na interface, fazendo com que as informações extras diminuam a visibilidade de informações relevantes na interface do aplicativo. Os designers devem ter cuidado ao exibir informações na tela. Além disso, interfaces sobrecarregadas podem gerar estresse para o usuário e também podem aumentar o consumo de recursos do aplicativo em um smartphone.

Benefícios: Se o aplicativo utilize um design minimalista, o dispositivo usa menos recursos, o que deve levar a um melhor desempenho e também minimiza a quantidade de informações visuais do usuário que podem levar a menos estresse e exaustão.

Problemas: Para distinguir entre um design minimalista e uma interface sobrecarregada, é necessário que o avaliador defina um critério subjetivo no momento da avaliação. Se o avaliador enfrentar uma interface sobrecarregada, há claramente um problema relacionado a essa heurística.

Sub-heurísticas:

120. O uso de imagens ou ícones é utilizável prioritariamente a textos na aplicação?
121. Somente as informações essenciais para a tomada de decisões são exibidas na tela?
122. Todos os ícones de um conjunto são visuais e conceitualmente distintos?
123. Objetos grandes, linhas arrojadas e áreas simples foram usadas para distinguir ícones?
124. Os grupos significativos de itens são separados por espaços em branco ou vazios?
125. Cada tela de entrada de dados tem um título curto, simples, claro e distinto?
126. Os rótulos de *labels* são breves, familiares e descritivos?
127. As solicitações são expressas afirmativamente e usam a voz ativa?
128. Cada opção de menu de nível inferior é associada a apenas um menu de nível superior?

129. Os títulos dos menus ou itens da interface são breves, mas longos o suficiente para se comunicar?
130. O sistema não usa muitos tipos de letra (fontes podem ser usadas para enfatizar o conteúdo, mas muitas fontes podem confundir os usuários)?
131. Os elementos de informação (por exemplo, imagens, ícones) se destacam do fundo?
132. Animações em movimento desnecessárias de informações (por exemplo, *zoom in*, *zoom out*) são evitadas?

2.3.10 HU10 – Ajuda ao Usuário para Reconhecer, Diagnosticar e Recuperar-se de Erros

Definição: O aplicativo deve exibir mensagens de erro em uma linguagem familiar para o usuário, indicando com precisão o problema e sugerindo uma solução construtiva.

Explicação: Quando ocorre um erro, o usuário não precisa de detalhes técnicos ou mensagens de alerta criptográficas, mas sim de mensagens claras em uma linguagem reconhecível, com instruções sobre como se recuperar do erro. Se possível, o aplicativo deve sugerir construtivamente uma solução (que também pode incluir dicas, perguntas frequentes, etc.). Se não houver solução para o erro, ou se o erro tiver um efeito insignificante, o aplicativo deve permitir que o usuário lide com o erro normalmente.

Benefícios: Sendo capaz de obter mensagens de ajuda e instruções sobre como se recuperar do erro, os usuários são ajudados a reduzir sua frustração ao lidar com o erro.

Problemas: O avaliador deve distinguir entre esta heurística e “HU5 – Prevenção de Erros”. A principal diferença aqui é o tempo. Se o erro ainda não aconteceu, estamos falando de prevenção (HU5), caso contrário, pode ser um problema relacionado a essa heurística.

Sub-heurísticas:

133. O som ou algum *feedback* tátil são usados para sinalizar um erro?
134. As solicitações são declaradas construtivamente, sem críticas explícitas ou implícitas ao usuário?
135. As mensagens de erro são breves e não ambíguas?
136. As mensagens de erro são redigidas para que o sistema, não o usuário, assuma a culpa?
137. Se forem usadas mensagens de erro bem-humoradas, elas são apropriadas e inofensivas para a população de usuários?

138. As mensagens de erro estão gramaticalmente corretas?
139. As mensagens de erro evitam o uso de pontos de exclamação?
140. Mensagens de erro evitam o uso de palavras violentas ou hostis?
141. As mensagens de erro evitam um tom antropomórfico?
142. Mensagens de erro no sistema usam um estilo gramatical, forma, terminologia e abreviações consistentes?
143. As mensagens colocam os usuários no controle do sistema?
144. A linguagem de comando usa a sintaxe normal do objeto de ação?
145. Se um erro for detectado em um campo de entrada de dados, o sistema realça o erro e/ou o elemento que precisa ser alterado?
146. As mensagens de erro informam o usuário sobre a gravidade do erro?
147. Mensagens de erro sugerem a causa do problema?
148. Mensagens de erro fornecem informações semânticas apropriadas?
149. Mensagens de erro indicam qual ação o usuário precisa executar para corrigir o erro?
150. Se o sistema oferecer suporte a usuários iniciantes e especialistas, vários níveis de detalhes da mensagem de erro estarão disponíveis?

2.3.11 HU11 – Ajuda e Documentação

Definição: O aplicativo deve fornecer documentação e ajuda fáceis de se encontrar, centrado na tarefa atual do usuário, e indicar passos concretos a serem seguidos para execução das funcionalidades do aplicativo.

Explicação: O aplicativo deve fornecer acesso a informações detalhadas sobre as funcionalidades disponíveis de maneira clara e simples, a partir de qualquer parte ou estado do sistema onde o usuário está localizado. É recomendado que estas informações estejam incluídas no aplicativo e facilmente acessíveis, ou de outra forma, que a documentação esteja disponível em um site. Para interações muito complexas ou difíceis para os usuários iniciantes entenderem, como gestos multitoque em um *smartphone touchscreen* que podem ser difíceis, neste caso é aconselhável apresentar tutoriais, documentação e ajuda para que os usuários sejam elucidados sobre a forma correta de execução do gesto e assim o usuário atinge seu objetivo no uso de uma certa funcionalidade do aplicativo.

Benefícios: Por meio do maior conhecimento do sistema, o usuário tende a cometer um menor número de erros e a ter uma maior eficiência no uso do aplicativo, bem como,

pode obter um melhor conhecimento do sistema através de uma documentação rica e também reduzir o risco de cometer erros.

Problemas: A principal dificuldade em aplicar esta heurística é como diferenciá-la de “HU5 – Prevenção de Erros”. Embora a documentação e as mensagens de ajuda possam evitar erros, essa heurística é muito focada em instruções sobre como usar o sistema (como interação por gestos do usuário), informações adicionais sobre opções e configuração, e assim por diante.

Sub-heurísticas:

151. Caso existam instruções/tutoriais de ajuda, eles são visualmente distintos e visíveis do resto da interface?
152. As instruções seguem a sequência de ações do usuário?
153. Se os itens do menu ou lista são ambíguos, o sistema fornece informações adicionais quando um item é selecionado?
154. Navegação: a informação é fácil de encontrar?
155. Apresentação: o layout é bem projetado?
156. Conversação: as informações são precisas, completas, e fáceis de entender?
157. As informações são relevantes?
 - Orientada por metas (o que posso fazer com este programa?);
 - Descritivo (para que serve isso?);
 - Procedural (como faço essa tarefa?);
 - Interpretativo (por que isso aconteceu?);
 - Navegação (onde estou?).
158. Existe ajuda para dados sensíveis do usuário?
159. O usuário pode alterar o nível de detalhe (da informação) disponível?
160. É fácil acessar e retornar do sistema de ajuda?
161. Os usuários podem retomar o trabalho de onde pararam depois de acessar a ajuda?
162. Se uma seção de FAQ existir, a seleção e a texto das perguntas e respostas estão relacionadas corretamente com o assunto?
163. Quando os usuários começam a usar o sistema pela primeira vez, o sistema fornece instruções (ou dicas)?

164. Se forem fornecidas instruções para o usuário da primeira vez, elas podem ser caracterizadas com o descrito abaixo?

- Ser simples e claro;
- Focando em alguns recursos (por exemplo, recursos usados com frequência);
- Sendo necessário que os usuários iniciem.

2.3.12 HU12 – Interação Agradável e Respeitosa com o Usuário

Definição: O dispositivo deve fornecer uma interação agradável com o usuário, de forma com que este não se sinta desconfortável ao utilizar a aplicação.

Explicação: Os aplicativos móveis são projetados para serem utilizados em dispositivos portáteis (smartphones). Deste ponto de vista, a ergonomia e o conforto desempenham um papel muito importante na interação entre o usuário e o aplicativo. Os botões devem ser colocados em posições reconhecíveis pelo usuário com base em sua experiência com outros aplicativos similares. Além disso, é necessário analisar o contexto de uso no qual o aplicativo será inserido, como uma aplicação para corredores que será utilizada durante a prática esportiva, deve haver uma interface totalmente inclinada para o conforto do usuário e ergonomia durante o uso da aplicação relativo à disposição dos elementos. A interface como um todo tem que ser focada nas características de seu público alvo, e pode incluir usuários com características gerais e/ou também os mais específicos, como usuários idosos que tendem a ter certas características especiais (acessibilidade), a título de exemplo: mudanças psicossociais e funcionais que afetam a visão, a audição, o movimento, a cognição e sua relação consigo mesmas e com os outros ao seu redor, tais fatores influenciam a maneira como os elementos são organizados na interface e como o usuário irá interagir com o sistema.

Benefícios: O aplicativo se torna mais intuitivo usando elementos de layout na interface que são familiares ao usuário, como colocar botões em posições reconhecíveis ou ao variar a orientação da aplicação entre vertical e horizontal, mantendo o mesmo layout dos elementos na tela. Adaptar toda a interface do aplicativo, levando em consideração as características do usuário e o contexto de uso, torna o usuário final mais eficiente nas interações com a aplicação.

Problemas: Um possível problema na aplicação desta heurística pode ser o caso de usuários com necessidades/características especiais. Neste caso, o avaliador deve usar seu próprio critério. Existem aplicativos que podem ser desconfortáveis para a maioria dos usuários e esse é um problema relacionado a essa heurística. Um aspecto muito importante a considerar é “popularidade”, ou seja, o que melhor se adéqua à maioria dos usuários.

No entanto, cada caso deve ser analisado separadamente e o avaliador deve estabelecer seus critérios.

Sub-heurísticas:

165. Os campos para entrada de dados que requerem digitação são mínimos?
166. Para telas de entrada de dados com muitos campos ou informações incompletas para preencher, os usuários podem salvar uma tela (ou parte dela) parcialmente preenchida?
167. As famílias tipográficas utilizadas no sistema são adequadas para leitura (não contém homógrafos, por exemplo, 1, l e L; Zero e O)?
168. O aplicativo implementa algum sistema de acessibilidade próprio ou suporta o da plataforma (por exemplo, voice over no sistema iOS utilizado para leitura e navegação por usuários com deficiência visual)?
169. O aplicativo é otimizado para o contexto de uso que usuários estão inseridos (por exemplo, usuário com deficiência motora)?
170. A disposição dos elementos de layout da aplicação é baseada e otimizada para o contexto em que o público alvo está inserido (por exemplo, aplicativo de mapas que geralmente é utilizado enquanto o usuário realiza outra atividade paralela)?
171. O sistema suporta ambas as orientações (horizontal e vertical)?
172. Caso o aplicativo funcione na orientação vertical e horizontal, a disposição dos elementos de layout se mantém os mesmos?
173. Cada ícone individual é um membro harmonioso de uma família de ícones?
174. Os detalhes excessivos no design do ícone foram evitados?
175. A cor foi usada com discriminação?
176. A cor foi usada especificamente para chamar a atenção, comunicar a organização, indicar mudanças de status e estabelecer relações?

2.3.13 HU13 – Privacidade

Definição: O aplicativo deve proteger os dados confidenciais do usuário.

Explicação: O aplicativo deverá solicitar a senha do usuário para a modificação de dados importantes, além de, prover informações sobre como os dados pessoais do usuário são protegidos e sobre conteúdos de direitos autorais. Áreas da aplicação que lidam com os

dados sensíveis do usuário devem conter autenticação para que qualquer tipo de interação ocorra.

Benefícios: O usuário obterá uma maior satisfação e confiabilidade no aplicativo ao saber que os seus dados pessoais estão protegidos.

Problemas: O objetivo aqui não é apenas a documentação ou maneiras de informar ao usuário em relação aos métodos de proteção utilizados, mas de fato ter recursos do aplicativo que garantam a privacidade dos dados do usuário (como exibições de dados diferenciadas o usuário e os seus seguidores, em um aplicativo de rede social).

Sub-heurísticas:

177. As áreas protegidas são completamente inacessíveis?
178. Áreas protegidas ou confidenciais tem a opção de serem acessadas com senhas?
179. O recurso de privacidade é eficaz e bem-sucedido na aplicação?
180. Para aplicativos multiusuários, o sistema evita que o usuário fique permanentemente conectado/logado?
181. Se o aplicativo armazenar informações de cartão de crédito, ele permite que os usuários decidam se desejam permanecer logados?
182. Há informações de como os dados pessoais são protegidos e sobre o *copyright* do aplicativo?
183. Caso exista transações no aplicativo, há a possibilidade de os usuários salvarem os números de confirmação para essa transação (por exemplo, enviando-os por e-mail)?

2.4 Trabalhos Correlatos

Foram encontradas na literatura diversas ferramentas que apoiam a análise de usabilidade. Entretanto, algumas ferramentas utilizam mecanismos diferentes em suas análises, como é o caso do trabalho proposto por Santos [17], que realizou uma análise através de uma ferramenta utilizando o contexto de uso do usuário. Portanto, foi realizado um filtro entre as ferramentas, trazendo apenas as ferramentas que fazem uso das heurísticas de usabilidade.

A ferramenta Heva [10] foi desenvolvida com o intuito de apoiar análises de usabilidade aplicadas às disciplinas de Interação Humano Computador (IHC), através das heurísticas de usabilidade. Ela possui o objetivo de realizar análises de usabilidade somente em sistemas Web, não sendo possível realizar análises no contexto de dispositivos móveis. A Heva foi desenvolvida com a linguagem de programação Java, e é uma extensão para

o navegador Mozilla Firefox, fazendo com que a ferramenta só possa ser usada nesse navegador.

A ferramenta THEM [18] é um sistema Web capaz de auxiliar qualquer análise de usabilidade baseada em heurísticas, independente do contexto. É uma ferramenta com muitas funcionalidades, sendo possível até mesmo criar grupos de especialistas para realizar as análises. Contudo, o fato de ser um sistema Web pode atrapalhar nas análises em dispositivos móveis, tendo em vista que são necessário dois dispositivos para a análise, sendo um dispositivo para acesso ao sistema e o próprio dispositivo móvel contendo a aplicação a ser analisada. Uma solução para esse problema seria o uso de emuladores de dispositivos móveis em computadores, porém essa abordagem pode comprometer a análise, pois perderia o contexto de uso de um dispositivo móvel.

O SUIT [19] possui um funcionamento semelhante ao THEM. Sendo um sistema Web, ele também realiza análises baseadas em heurísticas, independente do contexto. Possui um grande diferencial no aviso de notificações via e-mail, trazendo uma comodidade para gestores de equipes de especialistas. Em contrapartida, traz a desvantagem do uso de múltiplos dispositivos para a avaliação de aplicações em dispositivos móveis.

Assim como a ferramenta THEM e a ferramenta SUIT, a ferramenta Mockup DUE [20] é capaz de auxiliar análises de usabilidade em heurísticas, independente do contexto. Porém, diferentemente da THEM, foi desenvolvido uma aplicação Desktop, não sendo necessário o uso de nenhum navegador para a avaliação por meio dessa ferramenta. Entretanto, a ferramenta apresenta o problema de possuir pelo menos dois dispositivos para a análise, como um modo de não sacrificar o contexto de uso na análise de uma aplicação em um dispositivo móvel.

Há também a ferramenta MHET [21], um apoio ferramental nas análises de usabilidade baseadas em heurísticas. Trata-se de uma ferramenta voltada para dispositivos móveis com sistema operacional Android. Por meio do uso dessa ferramenta, o especialista é capaz de realizar análises de usabilidade em *websites* através do contexto de dispositivos móveis. Porém, a ferramenta se limita somente à *websites*, não sendo possível realizar uma análise de outras aplicações no dispositivo.

Diferentemente dos trabalhos anteriores, neste trabalho foi desenvolvida uma ferramenta voltada para a análise de usabilidade baseada nas heurísticas para dispositivos móveis. Conforme apresentado na Tabela 2.1, é possível observar que há uma ausência de uma ferramenta que apoia os especialistas em usabilidade durante a avaliação heurística de uma aplicação. Por mais que a ferramenta MHET seja executada em um dispositivo móvel, ela é capaz apenas de realizar análises em aplicações Web, não sendo possível analisar outras aplicações no próprio dispositivo. Além disso, a ferramenta proposta irá contemplar todas as heurísticas e sub-heurísticas propostas por Costa [8, 9].

Tabela 2.1: Comparativo entre as ferramentas existentes e a proposta neste trabalho

Parâmetros	Heva	THEM	SUIT	Mockup DUE	MHET	Tuhm
Foco de Inspeção	Aplicações Web	Independente	Aplicações Web	Independente	Aplicações Web	Aplicações Mobile
Método de Inspeção	Avaliação Heurística					
Geração de Relatórios	Sim	Sim	Sim	Sim	Sim	Sim
Multi-plataforma	Não	Sim	Sim	Sim	Não	Não
Suporte a Colaboração	Não	Sim	Não Informado	Sim	Não	Não

2.5 Síntese do Capítulo

Neste capítulo foram apresentados os conceitos de Usabilidade e Avaliação Heurística. Também foi apresentado o Conjunto de Heurísticas de Usabilidade proposto por Costa et al. [8, 9]. Além disso, foram apresentadas ferramentas desenvolvidas anteriormente, e o diferencial da ferramenta proposta neste trabalho. No Capítulo 3, serão apresentados mais detalhes a respeito da ferramenta desenvolvida, e uma descrição de como usá-la.

Capítulo 3

Desenvolvimento

Este Capítulo apresenta como a ferramenta proposta neste trabalho foi desenvolvida. Além disso, descrevemos as tecnologias e a arquitetura utilizadas no processo de desenvolvimento, bem como os seus requisitos. Também é apresentado o funcionamento interno da aplicação.

3.1 Requisitos do Sistema

A ferramenta proposta tem como objetivo principal auxiliar a avaliação de usabilidade baseada nas heurísticas para dispositivos móveis propostas por Costa et al. [8, 9]. As principais funcionalidades e a arquitetura foram definidas através da revisão de literatura, apresentada na Seção 2.4. A partir das informações coletadas, foi possível especificar os requisitos da ferramenta, que precisa contemplar as seguintes funcionalidades:

1. Gerenciamento de Avaliações;
2. Gerenciamento dos Problemas de Usabilidade;
3. Geração de Relatório da Avaliação de Usabilidade;
4. Compartilhamento do Relatório da Avaliação de Usabilidade.

No Gerenciamento de Avaliações, o especialista deve ser capaz de visualizar, cadastrar, editar e excluir avaliações da aplicação. Através da criação de avaliações, o especialista é capaz de vincular problemas encontrados à avaliação cadastrada. Além disso, a geração de relatórios é feita com base nos dados da avaliação e dos problemas vinculados a esta avaliação.

No Gerenciamento dos Problemas de Usabilidade, o especialista deve ser capaz de visualizar, cadastrar, editar e excluir problemas em uma avaliação. Os problemas cadastrados

são todos vinculados a uma avaliação. Para cada problema cadastrado, o especialista também deve informar as heurísticas que o problema violou e a gravidade do problema.

Na Geração de Relatório da Avaliação de Usabilidade, o especialista deve ser capaz de visualizar um relatório geral da avaliação selecionada. Esse relatório deve possuir estatísticas da avaliação, como quantidade de problemas encontrados e o número de heurísticas violadas. Também deve ser possível visualizar os detalhes de cada problema de usabilidade encontrado na avaliação.

No Compartilhamento do Relatório, o especialista deve conseguir compartilhar o relatório com outros especialistas, mesmo que eles não usem o aplicativo. Deve ser construído um documento para que qualquer especialista possa ter acesso às estatísticas geradas pelo relatório. Outro ponto importante é que o documento esteja em um formato compatível com impressão.

3.2 Tecnologia e Arquitetura

A ferramenta foi desenvolvida para ser executada no sistema operacional Android. O Android é um sistema operacional voltado para dispositivos móveis. Ele tem a vantagem de ser um sistema de código aberto, permitindo que qualquer pessoa tenha acesso ao seu código-fonte. Além disso, ele possui um ambiente de desenvolvimento próprio, facilitando o processo de fabricação da aplicação.

Através do ambiente de desenvolvimento próprio do Android, é possível desenvolver aplicações na linguagem de desenvolvimento Java e na linguagem Kotlin. Para o presente trabalho, foi adotada a linguagem Kotlin. A linguagem Kotlin é a linguagem de desenvolvimento oficial do Android [22], melhorando a integração com os recursos do sistema. Além disso, o Kotlin também traz a vantagem de tornar o código mais compacto e mais seguro quando comparado ao Java [23].

Foi utilizado o banco de dados SQLite, pois é o banco de dados padrão do Android [22]. Através do SQLite, é possível utilizar a biblioteca Room do Android [22]. A biblioteca Room realiza uma integração entre aplicativo e banco de dados em aplicações Android. Ela possui recursos que facilitam o desenvolvimento, além de tornar o código mais robusto e seguro.

A arquitetura utilizada no desenvolvimento foi a MVVM (Model-View-ViewModel). Essa é a arquitetura padrão de desenvolvimento Android [22], pois, além de dividir a aplicação em camadas, essa arquitetura não é afetada pelo ciclo de vida da aplicação. A principal diferença na MVVM está nas *ViewModels*, pois a Interface de Usuário (UI) anexa observadores nos dados da *ViewModel*, e sempre que uma alteração é realizada nesses dados, a *ViewModel* notifica seus observadores da alteração. A *ViewModel* também

é responsável por lidar com a lógica de negócio do sistema, e é ela quem realiza requisições para a camada *Repository*, para que o acesso ao banco de dados seja feito. É possível visualizar o diagrama da arquitetura na Figura 3.1.

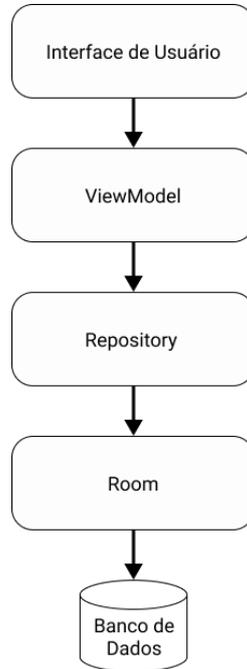


Figura 3.1: Diagrama da arquitetura do Tuhm.

Foi utilizada também a biblioteca Koin para a injeção de dependência. Através da injeção de dependência, é possível obter baixo acoplamento entre as camadas da arquitetura [24]. A biblioteca Koin é uma biblioteca própria para injeção de dependência na linguagem Kotlin, tem integração com o Android e facilita no desenvolvimento da injeção de dependências nos módulos.

3.3 Ferramenta Desenvolvida

A ferramenta Tuhm foi desenvolvida com o intuito de facilitar o trabalho da avaliação de aplicativos através das heurísticas de usabilidade. Essa ferramenta é capaz de gerenciar avaliações, podendo visualizar, cadastrar, editar e excluir as avaliações. Também é capaz de gerenciar os problemas de usabilidade encontrados em uma avaliação, podendo visualizar, cadastrar, editar e excluir esses problemas. Ao final do processo de avaliação, o usuário pode gerar um relatório com base na avaliação realizada, obtendo estatísticas e compartilhando um documento em PDF contendo o relatório. Sendo assim, a ferramenta atende aos requisitos especificados na Seção 3.1.



Figura 3.2: Tela inicial do Tuhm.

A Figura 3.2 apresenta a tela inicial do aplicativo. Essa tela é composta por uma listagem contendo todas as avaliações já cadastradas até o momento, conforme é mostrado no item 1. Ao clicar em uma das avaliações listadas, o usuário é direcionado para a tela de detalhes da avaliação, apresentada na Figura 3.3. Através do item 2, o usuário pode pesquisar entre as avaliações cadastradas, fazendo um filtro a partir do texto escrito pelo usuário. Clicando no item 3, o usuário é direcionado para a tela da Figura 3.4, onde ele pode iniciar uma nova avaliação.

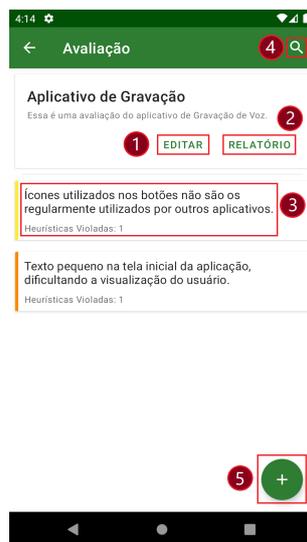


Figura 3.3: Tela de detalhes da avaliação, contendo a listagem dos problemas.

Na Figura 3.3 é possível ver a tela contendo os detalhes da avaliação e os problemas

já cadastrados até o momento. No botão do item 1, o usuário é direcionado para a tela de edição da avaliação, podendo editar os dados ou excluir uma avaliação, conforme Figura 3.4. Através do botão do item 2, o usuário vai para a tela de relatório da Figura 3.5, onde encontrará informações sobre a avaliação selecionada. No item 3, é possível ver que há uma lista com os problemas já cadastrados. Cada item dessa lista possui uma borda colorida à esquerda, que indica a sua gravidade. Ao clicar em um item desta lista, o usuário é redirecionado para a tela de edição de problemas da Figura 3.6. Através do item 4, o usuário é capaz de filtrar os problemas contidos na lista. Já no botão do item 5, o usuário pode adicionar um novo problema para a avaliação, conforme Figura 3.6.

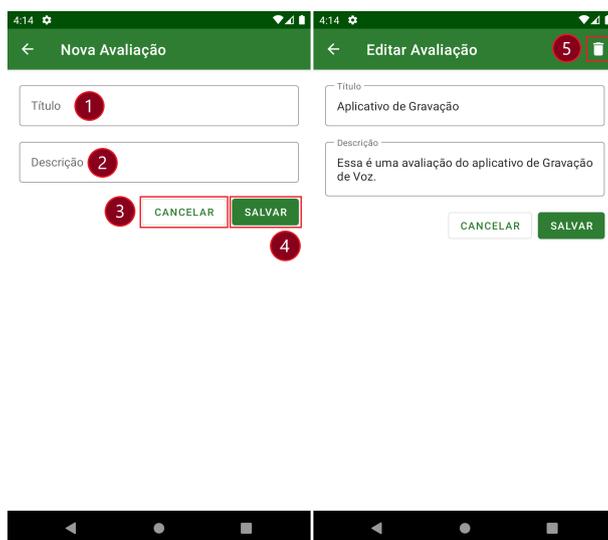


Figura 3.4: Tela de Adicionar/Editar uma Avaliação.

Na Figura 3.4 são apresentadas as telas de adicionar e editar uma avaliação. Nessa tela o usuário pode inserir um título e uma descrição contendo maiores detalhes sobre a avaliação a ser realizada, conforme itens 1 e 2. Para salvar a avaliação, basta que o usuário clique no botão do item 4, enquanto o botão do item 3 descarta qualquer alteração que o usuário tenha feito nesta tela. Através do item 5, o usuário é capaz de excluir a avaliação. É importante notar que o item 5 só aparece na tela de edição, pois só é possível excluir uma avaliação que tenha sido criada anteriormente.

A Figura 3.6 apresenta as telas para adicionar e editar um problema da avaliação. No botão do item 1, o usuário pode adicionar uma imagem ao problema, sendo possível editar a imagem (desenhar, cortar e escrever). No item 2 o usuário pode inserir uma descrição com os detalhes do problema encontrado. Através do item 3, o usuário pode escolher uma gravidade entre as quais foram apresentadas na Seção 2.2. No botão do item 4, o usuário é direcionado para a tela da Figura 3.7, onde pode selecionar as heurísticas na qual o problema está violando. No botão do item 5, o usuário salva o problema com todas



Figura 3.5: Tela de relatório da avaliação.

as informações inseridas na tela. Caso o usuário deseje descartar as alterações, basta utilizar o botão do item 6. No item 7 é mostrado como uma imagem é apresentada ao ser adicionada ao problema. No item 8 é apresentada uma lista contendo as heurísticas que foram violadas nesse problema. Através do item 9 o usuário pode excluir o problema da avaliação. É importante notar que o item 9 só aparece na tela de edição, pois só é possível excluir um problema que tenha sido criado anteriormente.

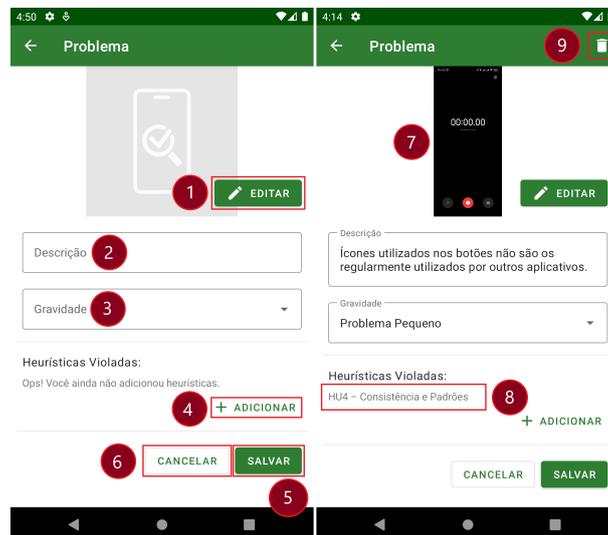


Figura 3.6: Tela de Adicionar/Editar um problema.

Na Figura 3.7 é apresentada a tela para selecionar as heurísticas que foram violadas em um determinado problema. Há uma listagem contendo as 13 heurísticas de Costa

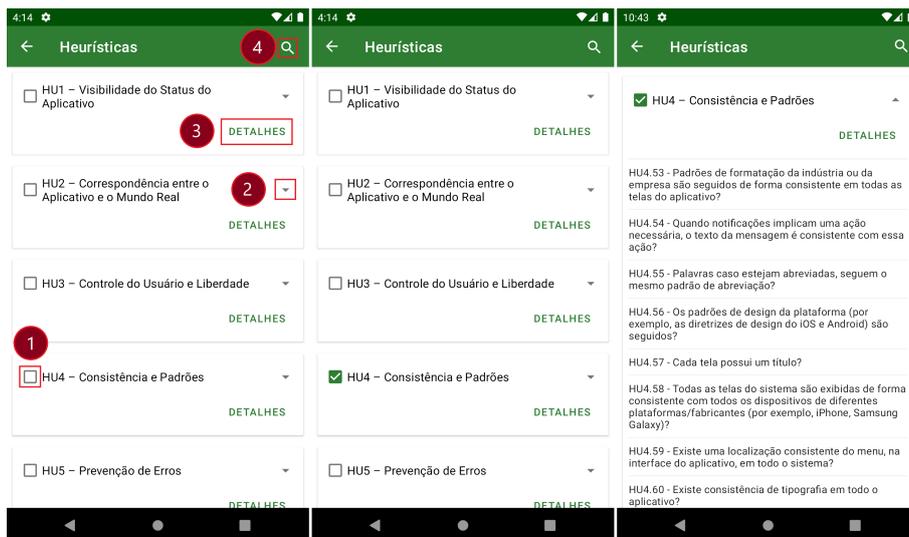


Figura 3.7: Tela de seleção de heurísticas violadas.

[8, 9] e as suas sub-heurísticas. Para selecionar uma heurística basta clicar em uma caixa de seleção, conforme o item 1. É possível ver as sub-heurísticas de uma determinada heurística clicando no ícone de seta, como mostrado no item 2. Clicando no botão do item 3 o usuário é direcionado para uma tela com os detalhes da heurística selecionada, contendo a definição, explicação, benefícios, problemas e suas sub-heurísticas, conforme Figura 3.8. No item 4 da Figura 3.7, o usuário é capaz de pesquisar entre as heurísticas e sub-heurísticas. Para finalizar o processo de seleção de heurísticas que o problema está violando, basta o usuário clicar no botão de voltar do dispositivo ou da tela, podendo continuar com o processo de avaliação.

A tela de detalhes de uma heurística é apresentada na Figura 3.8. Nessa tela é apresentado o título da heurística a ser detalhada, conforme item 1. No item 2 há a definição dessa heurística. No item 3 é possível ver a explicação para essa heurística. No item 4 há os benefícios de se utilizar essa heurística. No item 5 há os problemas dessa heurística, como os contras e as possíveis más interpretações. Já no item 6 é possível ver as sub-heurísticas dessa heurística.

Para a conclusão do processo de avaliação é gerado um relatório como o da Figura 3.5. Através da tela de relatório, é possível ver o nome e a descrição da avaliação no item 1. No item 2 é mostrado o total de problemas encontrados durante o processo de avaliação. Já o item 3 apresenta o número total de vezes em que as heurísticas foram violadas. No item 4 é apresentado um gráfico mostrando o número de problemas relacionado a cada nível de gravidade. Através do item 5, é apresentada uma listagem de problemas onde cada item da lista possui o início da descrição do problema, além de haver uma borda colorida à esquerda do item classificando sua gravidade. Os itens da lista de problemas

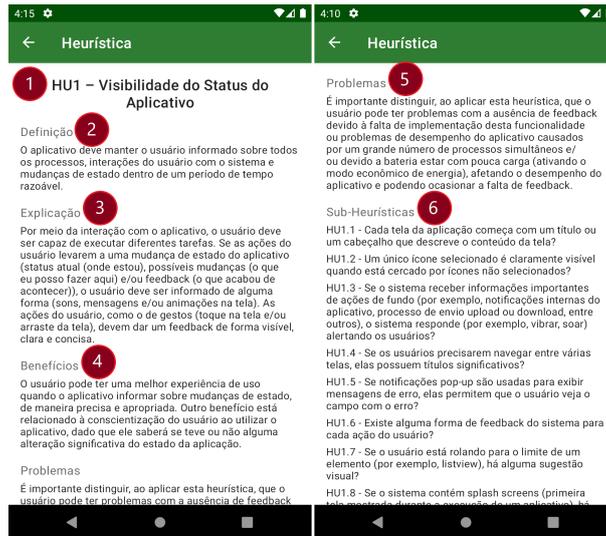


Figura 3.8: Tela de detalhes da heurística.

são classificados por nível de gravidade, indo do *Problema Catastrófico* ao *Inexistente*, dando prioridade aos problemas de maior gravidade. Clicando em um item desta lista de problemas, o usuário é direcionado para a tela contendo detalhes do problema encontrado, conforme Figura 3.9. O usuário, ao clicar no botão do item 6, também é capaz de gerar um relatório em formato PDF, como o da Figura 3.10, e compartilhar com outros usuários.



Figura 3.9: Tela de relatório dos detalhes do problema.

Na Figura 3.9 é apresentada a tela de relatório com os detalhes de um problema. Nesta tela há a imagem adicionada ao problema, conforme item 1. No item 2 é mostrada

a gravidade do problema. Já no item 3, há a descrição do problema encontrado. Enquanto o item 4 possui as heurísticas que foram violadas por esse problema.

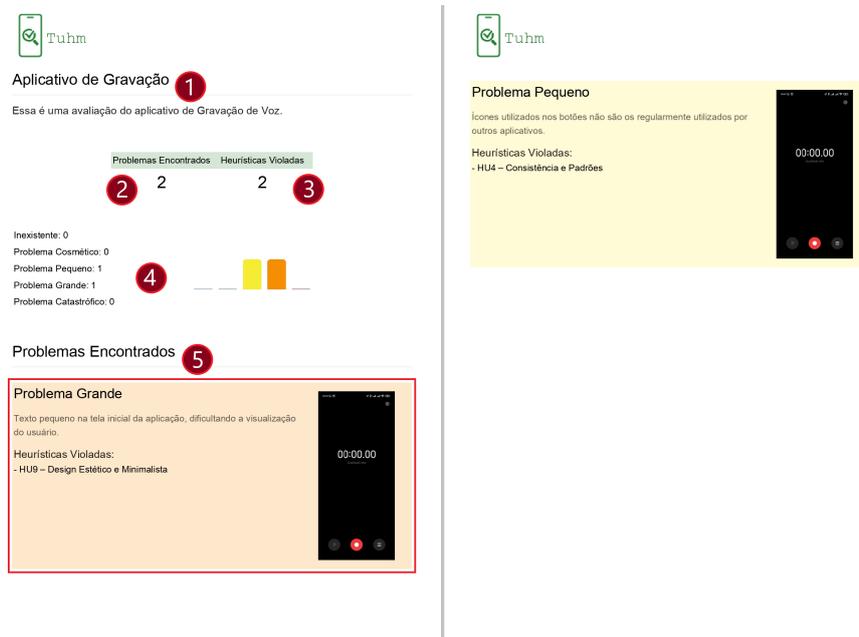


Figura 3.10: Arquivo PDF de um relatório.

Na Figura 3.10 é possível ver um relatório no formato PDF gerado pela aplicação. No item 1, o relatório apresenta o título e a descrição da avaliação. No item 2 é apresentado o número de problemas encontrados. Já no item 3, é mostrado o número de vezes em que heurísticas foram violadas. No item 4, é apresentado um gráfico mostrando o número de problemas relacionados a cada gravidade. Em seguida, no item 5, há uma lista com os detalhes de cada um dos problemas encontrados. Nesta listagem, para cada problema apresentado, há a gravidade do problema, a descrição, as heurísticas violadas e a imagem do problema.

3.4 Síntese do Capítulo

Neste capítulo foram apresentadas as tecnologias e a arquitetura utilizadas para o desenvolvimento da ferramenta. Também foram apresentados os requisitos que a ferramenta deveria possuir. Ao final, foi mostrada a ferramenta contendo imagens das telas e seu funcionamento. No próximo capítulo será apresentado um estudo de caso realizado com a ferramenta.

Capítulo 4

Estudo de Caso Prático

Neste capítulo são apresentados os resultados obtidos através da ferramenta desenvolvida no Capítulo 3, apresentando também as ameaças e as limitações da ferramenta. O estudo de caso prático foi realizado a partir da avaliação do aplicativo Escritório Social Virtual (ESV) [25, 26].

4.1 Avaliação Heurística do ESV

Para a realização da avaliação heurística utilizando a ferramenta proposta foi escolhido o aplicativo Escritório Social Virtual (ESV) [25, 26]. O aplicativo ESV é um aplicativo desenvolvido pela Universidade de Brasília (UnB) em parceria com o Governo do Distrito Federal (GDF) através da Fundação de Apoio à Pesquisa do Distrito Federal (FAP-DF) e do Conselho Nacional de Justiça (CNJ). O aplicativo tem o objetivo de complementar o atendimento presencial realizado nos escritórios sociais físicos do CNJ, prestando atendimento e serviço para as pessoas egressas do sistema penitenciário e seus familiares.

Através do aplicativo ESV, o usuário pode obter informações sobre serviços e políticas de atendimento às pessoas egressas do sistema prisional, informando também sobre oportunidades de ingresso em cursos, dicas de trabalho, entre várias outras funcionalidades [26]. O aplicativo foi lançado em abril de 2021. Atualmente, o app possui mais de mil downloads feitos na loja, estando disponível na Play Store (Android) e na App Store (iOS).

4.1.1 Procedimentos para a Avaliação Heurística

Durante o processo de seleção dos especialistas em usabilidade, foram selecionados especialistas com pelo menos 2 anos de experiência em usabilidade. Foi realizado contato com diversos especialistas através da rede social LinkedIn. Foi escolhido o LinkedIn por se

tratar de uma rede social com caráter mais profissional, sendo possível encontrar muitos especialistas em usabilidade. No total, 131 especialistas em usabilidade foram contactados, dos quais apenas 53 responderam ao contato. Entre os 53 especialistas, somente 17 demonstraram interesse em participar da pesquisa, e apenas 5 concluíram a avaliação.

Os especialistas foram separados entre especialistas que realizaram a avaliação utilizando a ferramenta e especialistas que realizaram a avaliação de forma manual, sem o uso da ferramenta. Através dessa divisão, é possível realizar uma comparação entre o uso da ferramenta e o método de avaliação tradicional. O perfil dos especialistas é apresentado na Tabela 4.1.

Tabela 4.1: Perfil dos Especialistas da Avaliação Heurística

ID	Posição	Local de Trabalho	Nível Acadêmico	Tempo de Experiência	Tipo de Avaliação
E1	Diretora de UX	Hu. Innovation	Graduada	9 anos	Avaliação com a Ferramenta
E2	Desenvolvedor iOS	Zup Innovation	Mestrado	2 anos	Avaliação com a Ferramenta
E3	UX Designer	Localiza	Pós-Graduado	10 anos	Avaliação com a Ferramenta
E4	UI/UX Designer	Fóton Informática	Pós-Graduado	10 anos	Avaliação Manual
E5	UX Designer	Sitel	Pós-Graduado	5 anos	Avaliação Manual

Para os especialistas que realizaram a avaliação com a ferramenta, foram enviados dois documentos. Em um documento foi disponibilizado as instruções para a realização da avaliação heurística. Esse documento foi dividido em 4 partes, sendo a primeira parte uma visão geral sobre como realizar a avaliação, a segunda parte apresenta as heurísticas de Costa [8, 9], na terceira parte há uma descrição de como usar a ferramenta e na quarta parte há a descrição de como a avaliação deve ser conduzida. No outro documento há um questionário para que o especialista possa responder após a avaliação. O questionário possui as seguintes questões:

1. Q.1: Qual o seu nível de satisfação com o processo utilizado? (1 para muito insatisfeito, 10 para muito satisfeito)
2. Q.2: O uso de uma ferramenta contribuiu no processo de avaliação?
3. Q.3: Há alguma sugestão ou melhoria a ser feita?

Para os especialistas que realizaram a avaliação sem a ferramenta, de forma manual, foram enviados três documentos. O primeiro documento possui as instruções para realização da avaliação heurística, sendo dividido em 3 partes. A primeira parte possui uma visão

geral sobre como realizar a avaliação heurística, a segunda parte apresenta as heurísticas de Costa [8, 9] e na terceira parte há a descrição da condução da avaliação. O segundo documento é um relatório para que o especialista possa inserir os problemas encontrados durante o processo de avaliação. Já no terceiro documento há um questionário para que o especialista responda após a avaliação. O questionário possui as seguintes questões:

1. Q.1: Qual o seu nível de satisfação com o processo utilizado? (1 para muito insatisfeito, 10 para muito satisfeito)
2. Q.2: Acredita que o uso de uma ferramenta iria contribuir no processo de avaliação?
3. Q.3: Há alguma sugestão ou melhoria a ser feita?

Entre os especialistas que realizaram a análise com a ferramenta e sem a ferramenta, somente a questão Q.2 do questionário é diferente. Isso se deve ao fato de que esta questão se refere à ferramenta utilizada, portanto, sendo designada para os especialistas que utilizaram a ferramenta. Já os especialistas que realizaram a avaliação de maneira manual, é utilizada uma pergunta a respeito da opinião do especialista sobre o uso de uma possível ferramenta, pois esses especialistas realizam a avaliação sem o uso da ferramenta.

4.1.2 Resultados das Avaliações Heurísticas

Após a realização das avaliações heurísticas mencionadas na Subseção 4.1.1, os especialistas fizeram a devolutiva do questionário respondido e um relatório contendo todos os problemas encontrados durante o processo de avaliação. Para os especialistas que fizeram a avaliação sem o uso da ferramenta, o relatório foi disponibilizado através de um documento onde o especialista pudesse inserir os detalhes de cada problema encontrado, além das heurísticas violadas e a sua gravidade. Já para os especialistas que utilizaram a ferramenta, a própria ferramenta gerou um relatório da avaliação, que foi o relatório devolvido por esses especialistas.

No relatório, para cada problema encontrado, há uma descrição detalhada do problema, além das heurísticas violadas e da gravidade do problema. Além disso, para os especialistas que utilizaram a ferramenta, o relatório também possui o número total de problemas encontrados, o número de heurísticas violadas e um gráfico contendo o número de problemas encontrados para cada gravidade, conforme o relatório apresentado na Seção 3.3. Através do relatório, foi possível obter dados estatísticos das avaliações realizadas. Também foi possível obter possíveis melhorias para a ferramenta com o uso do questionário. Essas melhorias são apresentadas na Seção 4.2.

Na Figura 4.1, é possível observar o gráfico contendo o tempo gasto por cada especialista para realizar a avaliação. O especialista E1 levou 28 minutos, o especialista E2

levou 40 minutos, o especialista E3 levou 1 hora e 5 minutos, o especialista E4 levou 28 minutos e o especialista E5 levou 2 horas e 11 minutos. O tempo médio dos especialistas que utilizaram a ferramenta foi de 44 minutos, enquanto o tempo médio dos que não utilizaram a ferramenta foi de 1 hora e 19 minutos. Com isso, é possível notar que o uso de uma ferramenta traz mais agilidade na avaliação heurística.

Observando a Figura 4.1, é possível notar que os especialistas E3 e E5 foram os especialistas que levaram mais tempo para a realização da avaliação. Através da Tabela 4.1, que apresenta o perfil dos especialistas, pode ser visto que estes dois especialistas são os únicos que atuam como *UX Designer*, mesmo que haja outro especialista com uma posição semelhante, como o caso do E4 que possui o papel de *UI/UX Designer*. Há uma diferença entre os termos UI e UX. A principal diferença é que UI se refere à aspectos visuais, enquanto o UX se refere à aspectos de usabilidade [27]. Como os especialistas E3 e E5 são focados principalmente na parte de UX, acredita-se que estes dois profissionais tenham uma maior preocupação em suas avaliações, motivo que fez com que suas avaliações levassem mais tempo.

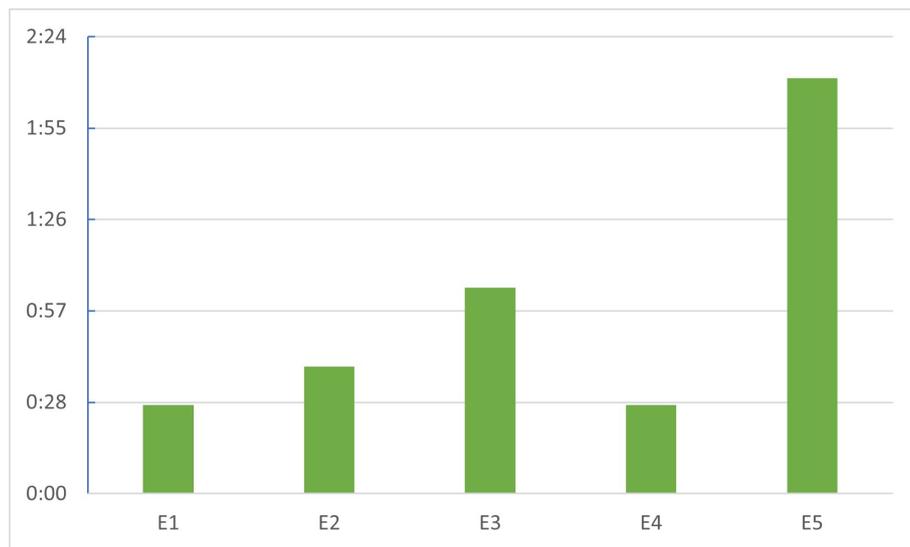


Figura 4.1: Tempo levado para conclusão da avaliação heurística para cada especialista.

Através da Figura 4.2, é possível observar o total de problemas encontrados para cada heurística. A heurística HU1 é a heurística que mais apresenta problemas, totalizando 11 problemas. Através da Subseção 2.3.1, é possível notar que a heurística HU1 trata-se de problemas de visibilidade de status do aplicativo. Como houve muitos problemas que violaram essa heurística, isso mostra que o aplicativo ESV possui problemas para informar ao usuário sobre o estado do aplicativo. Por exemplo, na tela de carregamento

e na reprodução de vídeos não há um componente visual que mostre ao usuário que o aplicativo ou um vídeo está sendo carregado.

Analisando o gráfico da Figura 4.2, é importante observar que as heurísticas HU10 e HU13 não tiveram problemas encontrados. Através da Subseção 2.3.10, é possível notar que a heurística HU10 trata de problemas de identificação e recuperação de erros. Como o aplicativo ESV tem uma proposta de apenas levar informações até o usuário, isso faz com que o usuário não seja levado a cometer algum erro durante a sua interação, fazendo com que o aplicativo não possua erros com esta heurística. Além disso, na Subseção 2.3.13 está a descrição da heurística HU13 que trata de problemas de privacidade do usuário. Como o aplicativo não possui uma área protegida e não necessita de um acesso autenticado do usuário, ele não precisa de uma proteção de dados confidenciais do usuário, pois o aplicativo não lida com esse tipo de dado, fazendo com que não haja problemas para essa heurística.

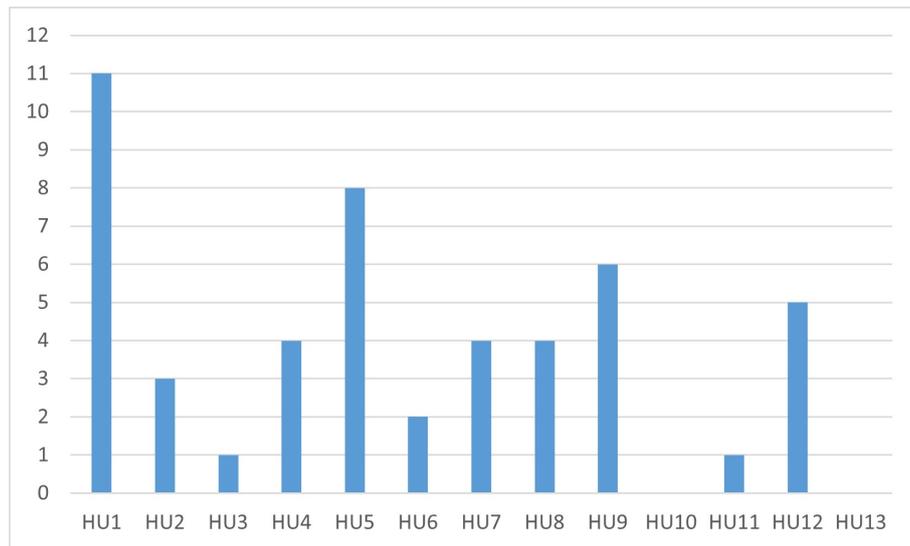


Figura 4.2: Número de problemas encontrados separados por heurística.

Na Figura 4.3 são apresentados os problemas encontrados por cada especialista divididos por gravidade. Através do uso da ferramenta, os especialistas descobriram em média 6,33 problemas cada, enquanto os especialistas que não utilizaram a ferramenta descobriram em média 13,5 problemas cada. Analisando as médias, os especialistas que não utilizaram a ferramenta foram capazes de descobrir mais problemas do que os especialistas que utilizaram a ferramenta.

Analisando os dados da Figura 4.3, é possível notar que os especialistas que não utilizaram a ferramenta encontram problemas com a gravidade *Inexistente*, que tem como descrição "Não concordo que seja um problema de usabilidade" conforme mostrado na

Seção 2.2. Problemas classificados com essa gravidade não são problemas de usabilidade, geralmente se trata de melhorias no design (UI), o que foge do objetivo desta pesquisa, que busca lidar apenas com problemas de usabilidade. Nesse quesito, a ferramenta se mostrou mais vantajosa, fazendo com que os especialistas tenham um maior foco na busca por problemas de usabilidade, pois os especialistas que utilizaram a ferramenta não tiveram problemas classificados com a gravidade *Inexistente*.

Entre os especialistas, o especialista E5 foi o único especialista que encontrou um problema com a gravidade *Problema Catastrófico*. Isso pode ter relação com o tempo gasto durante a realização da sua avaliação. Analisando o gráfico da Figura 4.1, é possível notar que o especialista E5 foi o especialista que levou mais tempo para a realização da sua avaliação. Levando à conclusão de que como o especialista passou mais tempo realizando a avaliação, isso o levou a encontrar problemas maiores. Além disso, através de um experimento realizado por Nielsen [16], foi descoberto que em um grupo de especialistas alguns problemas de usabilidade foram descobertos somente por um especialista, fato que se repete no presente trabalho.

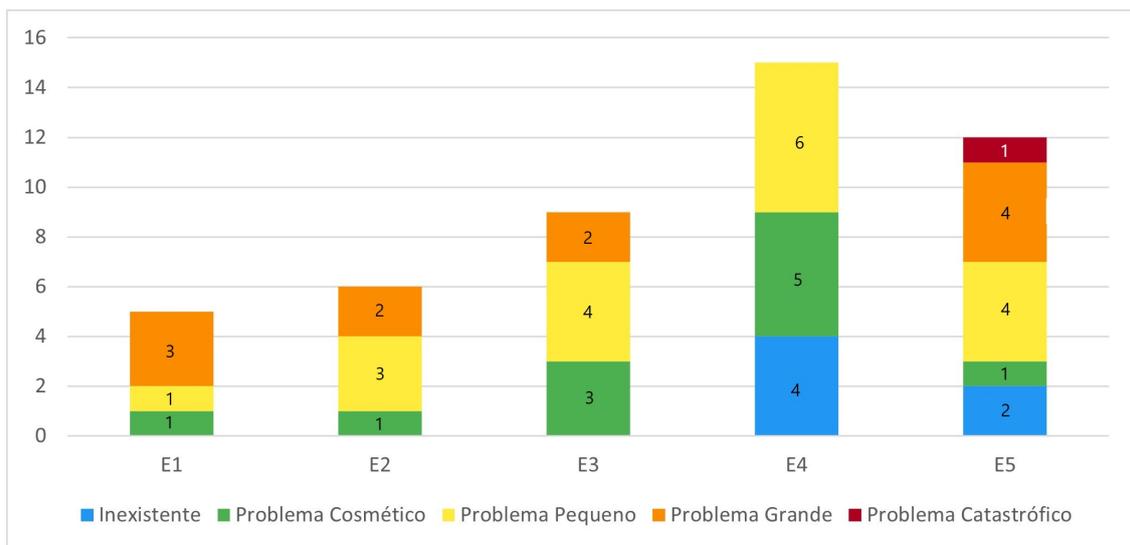


Figura 4.3: Número de problemas encontrados quanto a gravidade para cada especialista.

Por fim, ao finalizar a avaliação, os especialistas responderam um questionário que foi detalhado na Subseção 4.1.1, e em seguida, o questionário foi entregue pelos especialistas respondido. Para a questão Q.1, que diz: "Qual o seu nível de satisfação com o processo utilizado? (1 para muito insatisfeito, 10 para muito satisfeito)", os especialistas deram as seguintes respostas:

- E1: Nota 8. Usar a ferramenta foi muito satisfatório, no entanto, o material de explicação inicial, embora completo, era muito extenso;

- E2: Nota 8. No guia da avaliação poderia ter indicado em que momento entrar com os dados na ferramenta, para que alguns não percam tempo no início ou fim e fique padronizado o processo como um todo;
- E3: Nota 6;
- E4: Nota 7;
- E5: Nota 8.

Analisando as respostas para a questão Q.1, é possível notar que o nível de satisfação está bem próximo, tendo os especialistas que utilizaram a ferramenta uma média 7,33 e os especialistas que não utilizaram a ferramenta uma média 7,5. Entretanto, através dos comentários dos especialistas E1 e E2, é possível notar que a leitura do documento para a condução da avaliação, apresentado na Subseção 4.1.1, afetou negativamente a nota de satisfação.

A especialista E1 faz um comentário dizendo que o uso da ferramenta foi muito satisfatório, entretanto, diz que o material de explicação inicial era muito extenso. No material de explicação, que se trata do documento de condução da avaliação, há as heurísticas de usabilidade que foram apresentadas na Seção 2.3. Por se tratar de um conjunto de heurísticas muito grande contendo muitas informações, isso faz com que o documento de condução da avaliação se estenda muito, aumentando o tempo necessário para a leitura e a preparação da avaliação. Já o especialista E2 faz uma crítica ao guia da avaliação, que se trata do documento de condução da avaliação, pois ele encontrou dificuldade para saber quando utilizar a ferramenta. Através dessa crítica, é possível notar que a padronização da avaliação pode melhorar a experiência com a ferramenta, porque o usuário seria guiado para um uso mais eficaz.

A questão Q.2 foi diferente para os especialistas que utilizaram a ferramenta e os que não utilizaram, conforme mostrado na Subseção 4.1.1. Para os especialistas que utilizaram a ferramenta, a questão Q.2 diz: "O uso de uma ferramenta contribuiu no processo de avaliação?". Já para os especialistas que não utilizaram a ferramenta, a questão Q.2 diz: "Acredita que o uso de uma ferramenta iria contribuir no processo de avaliação?". Os especialistas deram as seguintes respostas:

- E1: Sim, contribuiu bastante principalmente por ter uma breve descrição das heurísticas e por ser um formato rápido de organização e categorização dos problemas encontrados;
- E2: Foi de fácil utilização, ajuda o avaliador a ter uma ferramenta rápida de entrada de dados para posterior utilização na avaliação dos dados obtidos;

- E3: Sim;
- E4: Uma busca dentro do aplicativo;
- E5: Sim. Boa parte dos elementos das interfaces podem ser perfeitamente avaliados por meio de um sistema, como exemplo: estados de botões, legibilidade dos textos e imagens, hierarquia semântica, uso acessível da paleta de cores, o que salvaria bastante tempo de trabalho. Entretanto, para a avaliação da efetividade da interpretação das informações na interface pelo usuário e o quanto o aplicativo está adequado ao conhecimento do usuário, uma análise sistêmica sem o *feedback* do próprio usuário pode tender ao erro.

Através das respostas para a questão Q.2, é possível notar que os especialistas que utilizaram a ferramenta deram uma resposta positiva, mostrando que a ferramenta ajudou a melhorar o processo de avaliação heurística. Além disso, os especialistas que não utilizaram a ferramenta forneceram comentários para a melhoria do processo de avaliação heurística.

A especialista E1 comenta que a ferramenta contribuiu bastante na avaliação. Os pontos destacados foram por a ferramenta possuir uma breve descrição das heurísticas e por ser um formato rápido de organização e categorização dos problemas encontrados. Já o especialista E2 diz que a ferramenta foi de fácil utilização, e que é uma forma rápida de entrada de dados, facilitando o processo de avaliação heurística. O especialista E3 não fez nenhum comentário.

O especialista E4 deu a sugestão de adicionar uma busca dentro do aplicativo, que seria a ferramenta proposta deste trabalho. Como não houve maiores detalhes, acredita-se que a busca na qual o especialista se refere seja uma busca de heurísticas que já foi implementada nesta ferramenta, conforme é mostrado na Figura 3.8. O especialista E5 cita algumas vantagens de se utilizar uma ferramenta. Entretanto, ele faz um comentário de que é necessário realizar uma análise junto ao usuário final, com o intuito de descobrir se o aplicativo avaliado está realmente adequado ao usuário.

Finalmente, na questão Q.3 que diz: "Há alguma sugestão ou melhoria a ser feita?", os especialistas deram as seguintes respostas:

- E1: Mudar a forma de apresentação do conteúdo que explica sobre o teste de usabilidade, fazendo o documento parecer menos extenso. As pessoas costumam compreender blocos pequenos por vez e aos poucos. Com isso, uma sugestão é começar com um pequeno mapa/guia do documento, exibindo os elementos em tópicos iniciais e com mais apelo visual. A partir daí, aumentar a navegação no documento. Quanto à ferramenta, minha única sugestão seria ter a possibilidade de anexar mais de uma

imagem em um mesmo problema, visto que às vezes isso se torna mais fácil de exibir os problemas de comportamento de fluxos;

- E2: Há somente algumas pequenas melhorias visuais que poderiam ser feitas, como colocar o *preview* da imagem selecionada na tela. A pesquisa das heurísticas quando não retornam nada, só fica o título geral da pesquisa, sem ficar claro que a pesquisa não retornou nada. Parece que foi feita a pesquisa de problemas cadastrados, e não dos problemas heurísticos cadastrados. Parece que as imagens, que eu selecionei do drive, não foram salvas e apresentadas no PDF, talvez seja um *bug*;
- E3: As Heurísticas estão muito detalhadas no documento. Poderia ter uma versão simplificada e depois aprofundar nos problemas;
- E4: Sugestões: Tiraria as abas; Colocaria uma busca na *home*; Na *home* faria um *header* com opção de hambúrguer à direita com localização, tela de parceiros, quem somos e informações sobre o aplicativo; Deixaria tudo com letras minúsculas e usaria nos textos mais tamanhos, pesos e cores para diferenciar as hierarquias por tipografia; Faria um alinhamento visual mais para a esquerda; O transformaria tudo em *cards* individuais e separaria os grupos visualmente com espaço em branco; Dentro do *card* colocaria o botão hambúrguer que viraria um *X* para voltar para a *home* (o voltar/nome da seção ficaria igual), criaria um padrão diferencial para dica, resumiria os textos, tiraria os textos com letras maiúsculas, deixaria mais arredondado os cantos de tudo (*card*/botão) e suavizaria as sombras;
- E5: Poderia ter sido apresentado um perfil de usuário para que eu tivesse melhores condições de avaliar aspectos mais subjetivos e relacionados à possibilidade de compreensão do usuário.

Os especialistas E1 e E3 fizeram um comentário sobre o tamanho do documento de condução da avaliação, que também foi discutido na resposta da especialista E1 para a questão Q.1. Esse problema do tamanho do documento foi causado pela descrição das heurísticas, pois se trata de um conjunto de heurísticas muito grande, conforme apresentadas na Seção 2.3. Contudo, a especialista E1 dá a sugestão de melhorar a apresentação do documento, mostrando uma necessidade de buscar meios de simplificar o uso das heurísticas de usabilidade durante uma avaliação. Analisando a resposta da especialista E1 para a questão Q.2, acredita-se que a ferramenta cumpriu este objetivo de simplificação das heurísticas, pois no seu comentário é dito que o uso de uma ferramenta contribuiu bastante no processo de avaliação, principalmente por ter uma breve descrição das heurísticas. Esta simplificação pode ser vista com mais detalhes nas Figuras 3.7 e 3.8.

Na resposta da questão Q.3, a especialista E1 sugeriu uma melhoria para permitir a adição de mais de uma imagem ao cadastrar um problema. Esta alteração traz um impacto muito grande na ferramenta, pois seria necessário alterar a estrutura do banco de dados, além de realizar alterações na estrutura de outras telas que apresentam essa imagem. Uma alternativa para um trabalho futuro seria a de adicionar uma tela de edição de imagens que permitiria ao especialista adicionar mais de uma imagem, fazendo com que ao final fosse gerada uma única imagem contendo as outras imagens que o especialista deseja adicionar.

O especialista E2 propôs quatro melhorias para a ferramenta:

1. A primeira melhoria foi a de adicionar a imagem do problema na tela que lista os problemas cadastrados, que foi apresentada na Figura 3.3. Esta alteração será implementada, pois ela traz uma facilidade para o especialista ao tentar encontrar um problema que ele já tenha cadastrado anteriormente;
2. A segunda melhoria foi a de informar ao especialista que a ferramenta não encontrou uma heurística com a pesquisa informada na tela da Figura 3.7. Esta melhoria será implementada, pois se trata de um erro de usabilidade na ferramenta que viola a heurística HU1 descrita na Subseção 2.3.1;
3. A terceira melhoria foi a de realizar a pesquisa de problemas cadastrados através das heurísticas que o problema viola. Esta melhoria não será implementada, pois, conforme é possível visualizar na Figura 3.3, o texto da heurística que o problema viola não é apresentado nessa tela, fazendo com que o especialista seja levado ao erro por apresentar resultados na pesquisa que não contém o texto que foi pesquisado;
4. A quarta melhoria foi a de permitir a adição de uma imagem ao problema a partir do aplicativo Google Drive. Esta melhoria será implementada, pois permite que o especialista insira uma imagem a partir de um outro aplicativo instalado no dispositivo, aumentando o número de opções na qual o especialista pode buscar a imagem a ser adicionada.

O especialista E4 fez apenas sugestões visuais para o aplicativo que foi avaliado. Como não houveram sugestões de melhorias para a ferramenta e nem sugestões para o processo de avaliação heurística, as sugestões deste especialista não serão discutidas, pois fogem do escopo deste trabalho.

O especialista E5 comentou que poderia ter sido apresentado um perfil do usuário para a realização da avaliação heurística. De fato, com um perfil de usuário o especialista é capaz de realizar avaliações mais eficazes, pois o especialista pode avaliar se o aplicativo está alinhado com as características destes usuários [28]. Além disso, há propostas de

conjuntos heurísticos que lidam com perfis de usuário específicos, como é o caso das heurísticas propostas por Silva et al. [5], que tem usuários idosos como público-alvo. Com isso, o especialista E5 mostra novamente a sua preocupação em introduzir o usuário no processo de avaliação, conforme sua resposta para a questão Q.2.

Através dos comentários para a questão Q.3, os especialistas sugeriram algumas melhorias para a ferramenta e foi discutido a viabilidade dessas melhorias. Essas melhorias estão contidas na Tabela 4.2, contendo o especialista que a sugeriu, a descrição da melhoria e se ela será implementada. Para as melhorias que serão implementadas, será realizado um refinamento na ferramenta para que essas melhorias sejam atendidas. Os detalhes desse refinamento são apresentados na Seção 4.2.

Tabela 4.2: Melhorias propostas pelos especialistas para a ferramenta

Especialista que sugeriu	Melhoria	Será implementado?
E1	Adicionar mais de uma imagem ao cadastrar um problema.	Não
E2	Adicionar uma visualização da imagem do problema na tela de lista de problemas.	Sim
E2	Adicionar um indicador informando que não houve resultados na pesquisa na tela de seleção de heurísticas.	Sim
E2	Buscar problemas cadastrados a partir do texto das heurísticas.	Não
E2	Adicionar suporte para a seleção de imagens para um problema a partir do aplicativo Google Drive.	Sim

4.2 Refinamentos da Ferramenta

Com base nos dados obtidos no questionário, apresentados na Tabela 4.2, foram realizados alguns refinamentos na ferramenta. Os refinamentos realizados, foram:

- Adicionar uma visualização da imagem do problema na tela de lista de problemas;
- Adicionar um indicador informando que não houve resultados na pesquisa na tela de seleção de heurísticas;
- Adicionar suporte para a seleção de imagens para um problema a partir do aplicativo Google Drive.

Na Figura 4.4, é possível observar a tela de detalhes da avaliação, previamente apresentada na Figura 3.3, após o refinamento. Não houve nenhum tipo de alteração funcional,



Figura 4.4: Tela de detalhes da avaliação após o refinamento, apresentando uma visualização da imagem do problema.

apenas alterações visuais. Através do item 1, pode-se notar que foi adicionada uma imagem para cada um dos problemas da listagem. Essa por sua vez se trata da imagem adicionada ao problema, facilitando a identificação dos problemas cadastrados. Para os problemas que não possuem uma imagem, o comportamento se mantém igual à Figura 3.3, mostrando apenas a descrição e o número de heurísticas violadas.

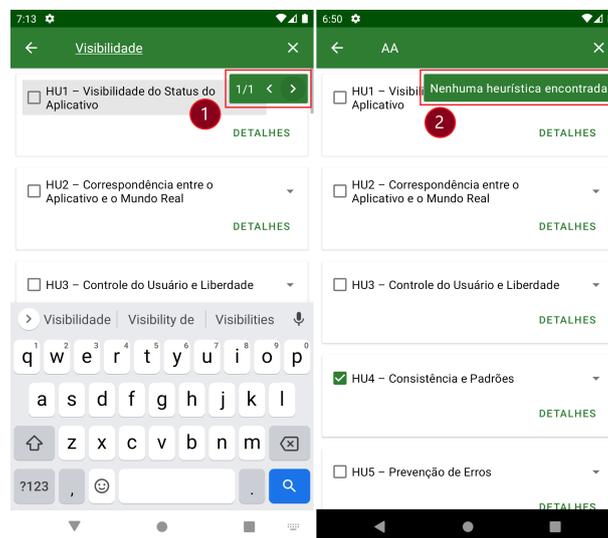


Figura 4.5: Tela de seleção de heurísticas violadas após o refinamento, apresentando indicador de que não houve resultados na pesquisa.

Através da Figura 4.5, é possível observar a tela de seleção de heurísticas após o refinamento, que foi apresentada anteriormente na Figura 3.7. Nessa tela foram realizadas

alterações tanto funcionais quanto visuais. No item 1, é possível ver um item para navegar entre os resultados da pesquisa. Neste item é mostrado qual o resultado está sendo apresentado entre o total de resultados encontrados, e há duas setas que possibilita ao usuário navegar entre os resultados da pesquisa. No item 2, é apresentado o cenário para quando não há resultados na pesquisa. Neste cenário, há uma mensagem informando ao usuário que nenhuma heurística foi encontrada a partir do texto informado no campo de pesquisa. Além disso, os botões de navegação são ocultados, impossibilitando ao usuário navegar entre as heurísticas através desse componente.

Na Figura 4.6, é apresentada a tela de adicionar e editar um problema após o refinamento, que foi apresentada anteriormente na Figura 3.6. Nessa tela houve apenas uma alteração funcional. Foi implementado um suporte para adicionar uma imagem ao problema a partir do aplicativo Google Drive. No item 1, é mostrada a tela do Google Drive, local onde a imagem está salva para ser adicionada ao problema. Já no item 2, é possível ver a imagem que foi selecionada no Google Drive adicionada ao problema.

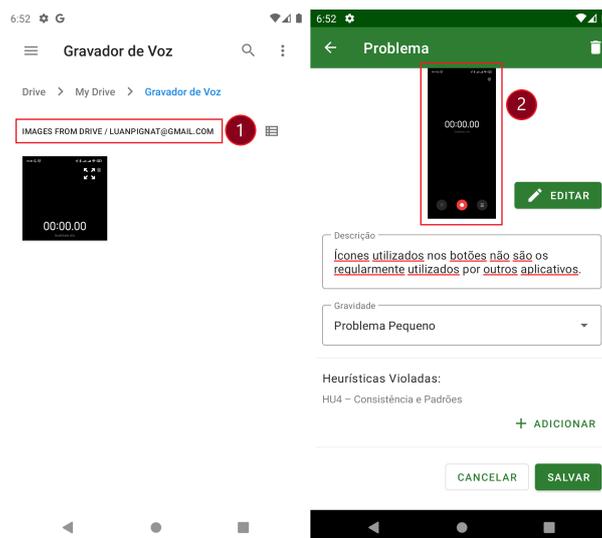


Figura 4.6: Tela de Adicionar/Editar um problema, com suporte para adição de imagens a partir do aplicativo Google Drive.

4.3 Riscos à Validação e Limitações da Ferramenta

Na Seção 4.1 foi apresentado um estudo de caso prático da ferramenta Tuhm, onde foram realizadas avaliações heurísticas do aplicativo Escritório Social Virtual (ESV) [25, 26]. Na Subseção 4.1.2 os resultados obtidos foram discutidos e com base nos dados coletados foram realizados refinamentos na ferramenta que são apresentados na Seção 4.2.

As avaliações foram conduzidas de forma remota, de tal modo que todos os documentos necessários para a realização da avaliação foram enviados aos especialistas e os especialistas que escolheram o local e o horário na qual gostariam de realizar a avaliação. Os detalhes da avaliação heurística foram apresentados na Seção 4.1. Como o ambiente não foi controlado, pode haver riscos quanto à validação da ferramenta, pois não houve um acompanhamento enquanto os especialistas realizavam as avaliações. Além disso, o número de especialistas também é um risco à validação, sendo necessário realizar mais avaliações com outros especialistas para que os dados tenham mais segurança, por uma questão de amostragem.

Entre as principais limitações da ferramenta estão o sistema operacional e o conjunto de heurísticas. Conforme visto na Seção 3.2, a ferramenta foi desenvolvida para o sistema operacional Android, podendo ser executada somente neste sistema. Entretanto, há outros sistemas operacionais disponíveis e bastante utilizados no mercado, como por exemplo o iOS da Apple. Além disso, a ferramenta utiliza somente as heurísticas de Costa [8, 9], limitando o especialista caso ele queira realizar uma avaliação utilizando um outro conjunto de heurísticas.

4.4 Síntese do Capítulo

Neste capítulo foi apresentado um estudo de caso prático da ferramenta Tuhm. Os resultados obtidos foram avaliados e discutidos, resultando em melhorias para a ferramenta. Em seguida, foram apresentados os riscos à validação e as limitações que a ferramenta possui. No próximo capítulo serão apresentadas as conclusões finais e os trabalhos futuros.

Capítulo 5

Considerações Finais

Neste capítulo é apresentada a conclusão do desenvolvimento e do uso da ferramenta Tuhm. Também são apresentados os trabalhos futuros.

5.1 Conclusão

Este trabalho apresentou a ferramenta Tuhm, uma ferramenta que auxilia no processo de avaliação heurística. Foi apresentado um comparativo com outras ferramentas, tendo como principal diferencial o fato de ser voltada para a avaliação de aplicativos em dispositivos móveis. Além disso, foi realizado um estudo de caso prático com o intuito de validar a ferramenta.

Primeiramente, foi feita uma busca por um conjunto heurístico com foco em aplicativos de dispositivos móveis. Foi encontrado o conjunto heurístico de Costa et al. [8, 9], apresentado na Seção 2.3. Em seguida, a ferramenta Tuhm foi desenvolvida e apresentada no Capítulo 3. Por fim, foi realizado um estudo de caso no Capítulo 4 onde especialistas realizaram uma avaliação de usabilidade. Os especialistas foram divididos de tal forma que alguns especialistas realizaram a avaliação utilizando a ferramenta e outros realizaram a avaliação sem a utilização da ferramenta. Através do estudo de caso, os especialistas produziram um relatório da avaliação, que são apresentados e discutidos no Capítulo 4. Com isso, os objetivos traçados na Seção 1.3 foram alcançados.

Através dos resultados apresentados na Subseção 4.1.2, foi possível notar que os especialistas que fizeram o uso da ferramenta precisaram de menos tempo para a realização da avaliação. Além disso, o uso de uma ferramenta tornou o processo de avaliação mais objetivo, pois os especialistas que utilizaram a ferramenta não encontraram problemas de gravidade *Inexistente*, que quer dizer "Não concordo que seja um problema de usabilidade", conforme apresentado na Seção 2.2, sendo que os especialistas que não utilizaram a ferramenta encontraram problemas com esta gravidade. Os especialistas que utilizaram a

ferramenta fizeram comentários positivos a respeito da mesma dizendo que esta contribuiu no processo de avaliação, sendo de fácil utilização e tendo um formato rápido de entrada de dados e uma organização e categorização dos problemas encontrados.

Analisando os resultados da Subseção 4.1.2, foi observado que os níveis de satisfação do processo de avaliação dos especialistas que utilizaram a ferramenta e dos que não utilizaram são semelhantes. Entretanto, o nível de satisfação foi afetado de forma negativa pois o documento de condução da avaliação é muito extenso, conforme apresentado na Subseção 4.1.1. Neste documento, há as heurísticas de usabilidade que foram apresentadas na Seção 2.3. Como o conjunto heurístico é muito grande, isso fez com que o documento se estendesse muito, aumentando o tempo necessário para a leitura e a preparação da avaliação. Com isso, houve uma necessidade de melhorar a apresentação dos conjuntos heurísticos, facilitando a leitura do especialista. Essa necessidade foi atendida na ferramenta Tuhm, havendo um modelo de apresentação mais simplificado para o conjunto heurístico, conforme discutido na Subseção 4.1.2.

Por fim, os especialistas sugeriram melhorias para a ferramenta através do questionário apresentado na Subseção 4.1.1. Essas sugestões foram apresentadas e discutidas na Subseção 4.1.2. Após uma discussão das melhorias sugeridas, foram realizados refinamentos na ferramenta com base nessas melhorias. Esses refinamentos são apresentados na Seção 4.2.

O processo de avaliação de usabilidade é bastante trabalhoso, sendo necessário buscar formas de tornar este processo mais produtivo, gerando resultados de forma mais rápida e de maneira mais satisfatória para o especialista. Com isso, este trabalho mostra a importância de ter uma ferramenta para o auxílio das avaliações heurísticas, pois através do uso de uma ferramenta, o especialista consegue realizar uma avaliação de forma mais produtiva e menos cansativa, como mostrado na Subseção 4.1.2. Portanto, este trabalho motiva o desenvolvimento de novas ferramentas, buscando melhorar o processo de avaliação heurística.

5.2 Trabalhos Futuros

Para trabalhos futuros, é interessante desenvolver ferramentas para outros sistemas operacionais, como por exemplo o iOS da Apple. Isso permite que mais especialistas possam utilizar essas ferramentas, além de facilitar a avaliação de aplicativos desenvolvidos para sistemas operacionais diferentes. Também é necessário ferramentas que utilizem mais conjuntos de heurísticas de usabilidade, ou uma plataforma na ferramenta que permita ao especialista cadastrar qualquer conjunto. Além disso, buscar maneiras de melhorar a

apresentação dos conjuntos heurísticos se faz necessário, tornando o processo de avaliação heurística menos moroso.

Como um trabalho futuro para a ferramenta Tuhm, seria interessante a implementação de um suporte à colaboração. Através do suporte à colaboração, vários especialistas poderiam realizar análises em um mesmo projeto. Isso facilitaria bastante o trabalho de avaliações heurísticas que são realizadas com um grupo de especialistas, podendo gerar um único relatório para várias análises.

Referências

- [1] Zeidler, Christian, Christian Kittl e Otto Petrovic: *An integrated product development process for mobile software*. Int. J. Mob. Commun., 6(3):345–356, 2008. <https://doi.org/10.1504/IJMC.2008.017515>. 1
- [2] Nielsen, Jakob e Rolf Molich: *Heuristic evaluation of user interfaces*. Em Chew, Jane Carrasco e John A. Whiteside (editores): *Conference on Human Factors in Computing Systems, CHI 1990, Seattle, WA, USA, April 1-5, 1990, Proceedings*, páginas 249–256. ACM, 1990. <https://doi.org/10.1145/97243.97281>. 1
- [3] Nielsen, Jakob: *Usability inspection methods*. Em Plaisant, Catherine (editor): *Conference on Human Factors in Computing Systems, CHI 1994, Boston, Massachusetts, USA, April 24-28, 1994, Conference Companion*, páginas 413–414. ACM, 1994. <https://doi.org/10.1145/259963.260531>. 1
- [4] Nielsen, Jakob: *Usability engineering*. Morgan Kaufmann, 1994. 1, 6, 8
- [5] Silva, Paula Alexandra, Kelly Holden e Phillip Jordan: *Towards a list of heuristics to evaluate smartphone apps targeted at older adults: A study with apps that aim at promoting health and well-being*. Em Bui, Tung X. e Ralph H. Sprague Jr. (editores): *48th Hawaii International Conference on System Sciences, HICSS 2015, Kauai, Hawaii, USA, January 5-8, 2015*, páginas 3237–3246. IEEE Computer Society, 2015. <https://doi.org/10.1109/HICSS.2015.390>. 1, 50
- [6] Lee, Young Seok, Sang W. Hong, Tonya L. Smith-Jackson, Maury A. Nussbaum e Kei Tomioka: *Systematic evaluation methodology for cell phone user interfaces*. Interact. Comput., 18(2):304–325, 2006. <https://doi.org/10.1016/j.intcom.2005.04.002>. 1
- [7] Bergman, Janne e Janne Vainio: *Interacting with the flow*. Em Sá, Marco de, Luís Carriço e Nuno Correia (editores): *Proceedings of the 12th Conference on Human-Computer Interaction with Mobile Devices and Services, Mobile HCI 2010, Lisbon, Portugal, September 7-10, 2010*, ACM International Conference Proceeding Series, páginas 249–252. ACM, 2010. <https://doi.org/10.1145/1851600.1851641>. 2, 7
- [8] Costa, Ruyther Parente da: *Conjunto de heurísticas de usabilidade para avaliação de aplicações móveis em smartphones*. Dissertation, supervisor prof. dra. edna dias canedo, Departamento de Ciência da Computação, Universidade de Brasília (UnB), 2019. 2, 8, 29, 30, 31, 37, 41, 42, 53, 54

- [9] Costa, Ruyther Parente da, Edna Dias Canedo, Rafael Timóteo de Sousa Júnior, Robson de Oliveira Albuquerque e Luis Javier García-Villalba: *Set of usability heuristics for quality assessment of mobile applications on smartphones*. IEEE Access, 7:116145–116161, 2019. 2, 8, 29, 30, 31, 37, 41, 42, 53, 54
- [10] Oeiras, Janne Yukiko Y., David Leonardo M. Bentolila e Mayara Costa Figueiredo: *Heva: uma ferramenta de suporte à avaliação heurística para sistemas web*. Em Filgueiras, Lucia e Marco Winckler (editores): *VIII Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC 2008), De 21 a 24 de Outubro 2008, Porto Alegre, ANAIS (Proceedings of the VIII Brazilian Symposium on Human Factors in Computing Systems 2008, Porto Alegre, RS, Brazil, October 21-24, 2008)*, volume 378 de *ACM International Conference Proceeding Series*, páginas 136–145. ACM, 2008. <https://dl.acm.org/citation.cfm?id=1497486>. 2, 28
- [11] Rivero, Luis e Tayana Conte: *Using the results from a systematic mapping extension to define a usability inspection method for web applications*. Em *Proceedings of the 24th International Conference on Software Engineering & Knowledge Engineering (SEKE'2012), Hotel Sofitel, Redwood City, San Francisco Bay, USA July 1-3, 2012*, páginas 582–587. Knowledge Systems Institute Graduate School, 2012. https://www.researchgate.net/publication/294580668_Using_the_Results_from_a_Systematic_Mapping_Extension_to_Define_a_Usability_Inspection_Method_for_Web_Applications. 2
- [12] Bento, António: *Como fazer uma revisão da literatura: Considerações teóricas e práticas*. Revista JA (Associação Académica da Universidade da Madeira), 7(65):42–44, 2012. 4
- [13] Harrison, Rachel, Derek Flood e David Duce: *Usability of mobile applications: literature review and rationale for a new usability model*. Journal of Interaction Science, 1(1):1–16, May 2013, ISSN 2194-0827. <https://doi.org/10.1186/2194-0827-1-1>. 7
- [14] Inostroza, Rodolfo, Cristian Rusu, Silvana Roncagliolo e Cristhy Jiménez Granizo: *Usability heuristics validation through empirical evidences: A touchscreen-based mobile devices proposal*. Em *31st International Conference of the Chilean Computer Science Society, SCCC 2012, Valparaíso, Chile, November 12-16, 2012*, páginas 60–68. IEEE Computer Society, 2012. <https://doi.org/10.1109/SCCC.2012.15>. 7, 8
- [15] Sagar, Kalpna e Anju Saha: *A systematic review of software usability studies*. International Journal of Information Technology, páginas 1–24, Dec 2017, ISSN 2511-2112. <https://doi.org/10.1007/s41870-017-0048-1>. 7
- [16] Nielsen, Jakob: *Finding usability problems through heuristic evaluation*. Em Bauersfeld, Penny, John Bennett e Gene Lynch (editores): *Conference on Human Factors in Computing Systems, CHI 1992, Monterey, CA, USA, May 3-7, 1992, Proceedings*, páginas 373–380. ACM, 1992. <https://doi.org/10.1145/142750.142834>. 8, 45

- [17] Santos, Rainara Maia: *Análise do contexto móvel nos testes de usabilidade de aplicações móveis*. Universidade Federal do Ceará, 2011. <http://www.repositorio.ufc.br/handle/riufc/25271>. 28
- [18] Santos, Francis, Glívia Barbosa, Ismael Silva e Flávio Coutinho: *Them: Ferramenta colaborativa para suporte a avaliações de interfaces baseadas na avaliação heurística*. Em *Anais do XIV Simpósio Brasileiro de Sistemas Colaborativos*, páginas 110–124, Porto Alegre, RS, Brasil, 2017. SBC. <https://sol.sbc.org.br/index.php/sbsc/article/view/9954>. 29
- [19] Ardito, Carmelo, Rosa Lanzilotti, Paolo Buono e Antonio Piccinno: *A tool to support usability inspection*. Em Celentano, Augusto (editor): *Proceedings of the working conference on Advanced visual interfaces, AVI 2006, Venezia, Italy, May 23-26, 2006*, páginas 278–281. ACM Press, 2006. <https://doi.org/10.1145/1133265.1133322>. 29
- [20] Rivero, Luis, Davi Viana e Tayana Conte: *Mockup due: Uma ferramenta de apoio ao processo de inspeção de usabilidade de mockups de aplicações web*. Em *IX Workshop Anual do MPS (WAMPS 2013)*, janeiro 2013. https://www.researchgate.net/publication/294580666_Mockup_DUE_Uma_Ferramenta_de_Apoio_ao_Processo_de_Inspecao_de_Usabilidade_de_Mockups_de_Aplicacoes_Web. 29
- [21] Patrono, Rafael Marcus da Costa: *Mhet: um aplicativo móvel para avaliação de websites utilizando o método de avaliação heurística*. Universidade Federal de Ouro Preto, 2019. <http://www.monografias.ufop.br/handle/35400000/2474>. 29
- [22] Google: *Android developers*. <https://developer.android.com/>, Acessado em 09/08/2021. 32
- [23] Ardito, Luca, Riccardo Coppola, Giovanni Malnati e Marco Torchiano: *Effectiveness of kotlin vs. java in android app development tasks*. *Inf. Softw. Technol.*, 127:106374, 2020. <https://doi.org/10.1016/j.infsof.2020.106374>. 32
- [24] Kocsis, Zoltan A. e Jerry Swan: *Dependency injection for programming by optimization*. CoRR, abs/1707.04016, 2017. <http://arxiv.org/abs/1707.04016>. 33
- [25] UnB: *Escritório social virtual*. <https://esv.bitbucket.io/>, Acessado em 09/09/2021. 40, 52
- [26] Mendonça, Walter Lucas Monteiro de, Pedro Henrique Teixeira Costa, Emille Catarine Rodrigues Cançado, Fernanda Lima, Edna Dias Canedo, Rodrigo Bonifácio e Luis Henrique Vieira Amaral: *From dusk till dawn: Reflections on the impact of COVID-19 on the development practices of a r & d project*. Em Cavalcante, Everton, Francisco Dantas e Thaís Batista (editores): *SBES '20: 34th Brazilian Symposium on Software Engineering, Natal, Brazil, October 19-23, 2020*, páginas 596–605. ACM, 2020. <https://doi.org/10.1145/3422392.3422446>. 40, 52
- [27] Kim, Sung Woo, Han Kyung Jo e Da Yun Ha: *Different ui, same UX: A design concept for implementing a locally-optimized and globally-unified user experience*. Em Marcus, Aaron (editor): *Design, User Experience, and Usability. Theory, Methods,*

Tools and Practice - First International Conference, DUXU 2011, Held as Part of HCI International 2011, Orlando, FL, USA, July 9-14, 2011, Proceedings, Part II, volume 6770 de *Lecture Notes in Computer Science*, páginas 440–448. Springer, 2011. https://doi.org/10.1007/978-3-642-21708-1_50. 43

- [28] Barbosa, Simone Diniz Junqueira, Bruno Santana da Silva, Milene Selbach Silveira, Isabela Gasparini, Ticianne Darin e Gabriel Diniz Junqueira Barbosa: *Interação Humano-Computador e Experiência do Usuário*. Autopublicação, 2021. 49