

Universidade Federal do Amazonas  
Instituto de Ciências Exatas  
Programa de Pós-Graduação em Informática

**Mandos: Um método de Interação com o usuário em  
Aplicativos Embarcados na Telefonia Móvel**

Mauro Ricardo da Silva Teófilo

Manaus - Amazonas  
Setembro de 2007

Mauro Ricardo da Silva Teófilo

**Mandos: Um método de Interação com o usuário em  
Aplicativos Embarcados na Telefonia Móvel**

Dissertação apresentada ao Programa de Pós-graduação em Informática da Universidade Federal do Amazonas, como requisito parcial à obtenção do título de Mestre em Informática. Área de concentração: Engenharia da Computação.

Orientador: Prof. Dr. Raimundo da Silva Barreto  
Co-Orientador: Prof. Dr. Ayres Mardem Almeida do Nascimento

Mauro Ricardo da Silva Teófilo

**Mandos: Um método de Interação com o usuário em  
Aplicativos Embarcados na Telefonia Móvel**

Dissertação apresentada ao Programa de Pós-graduação em Informática da Universidade Federal do Amazonas, como requisito parcial à obtenção do título de Mestre em Informática. Área de concentração: Engenharia da Computação.

Banca Examinadora

Prof. Dr. Raimundo da Silva Barreto – Orientador  
Departamento de Ciência da Computação – DCC/UFAM

Prof. Dr. Ayres Mardem Almeida do Nascimento – Co-orientador  
Departamento de Eletrônica e Automação – DET/UFAM

Prof. Dr. José Raimundo Gomes Pereira  
Departamento de Estatística – DE/UFAM

Prof. Dr. Alex Sandro Gomes  
Centro de Informática – CIn/UFPE

Manaus – Amazonas  
Setembro de 2007

*Dedico este trabalho aos meus pais Rocicler e João.*

# Agradecimentos

A Deus, a eterna inteligência do Universo, pela oportunidade de travar mais esta batalha.

À minha família que sempre me deu total apoio.

Ao meu professor orientador, Raimundo Barreto, pelo incentivo, paciência, disponibilidade e por dar um direcionamento sucinto para que este trabalho pudesse ser concluído com sucesso.

Ao meu professor co-orientador, Ayres Mardem, por auxiliar na fase inicial da formulação deste trabalho e principalmente por ter acreditado no meu potencial. Ao professor José Raimundo, que me auxiliou com comentários e orientações precisas sobre toda a parte estatística deste trabalho, sempre se colocando a disposição para solucionar qualquer dúvida.

Ao professor José Gabriel da Costa por me auxiliar a conceber a essência deste trabalho.

Ao professor Edson Gomi pela revisão inicial e comentários importantes sobre o texto que compõe este trabalho.

Ao professor Alex Sandro Gomes pelas excelentes observações e avaliação do presente trabalho, o que conduziu a um aprimoramento imprescindível do mesmo.

Ao grande incentivador e companheiro de profissão no INdT (Instituto Nokia de Tecnologia), Pedro Freitas, que sempre foi compreensivo e estimulador na realização deste trabalho e também a todos da equipe do INdT dentro os quais é válido citar o nome de Alan Castro.

A Glaucio Polzin, pelo apoio no manuseio do telefone celular, se mostrando sempre disposto a auxiliar.

A Fabiano Cruz, por ceder um aparelho celular BenQ para realização dos testes. A Ângelo Nicolay, pelo apoio inicial e liberação para o início do curso de mestrado e a todo o time da DBA Engenharia de Sistemas.

À Juliana Sena pelo apoio e revisão em alguns trechos da redação deste trabalho.

Meu agradecimento muito especial a todos os meus amigos que com muita boa vontade e simpatia opinaram e contribuíram para o resultado final deste trabalho, dentre as quais, cita-se, em ordem alfabética: Andréia Silva, Carol Benaion, Cláudio Chaparro, Cláudio Pinheiro, Cleiton Bittencourt, Eliana Kobata, Francisco Aguiar, Hugo Sena, Janaína Almeida, Jonildo Parente, Kelly dos Santos, Mário Teófilo, Mitsuyoshi Nishi, Nívia Teófilo, Rafael Doroteu, Rafael Vilardo, Susy Pinheiro e Talita Benaion.

À minha namorada Vitória Almeida pelo apoio, incentivo e carinho.

*A diferença básica entre um homem comum e um guerreiro é que um guerreiro toma tudo como desafio, enquanto um homem comum toma tudo como bênção ou como castigo.*

*Carlos Castaneda*

# Resumo

Com o uso intenso de aplicativos em aparelho celulares a questão da usabilidade passa a vigorar fortemente como objeto de estudo, sendo considerado um fator determinante do sucesso deste segmento da computação móvel. Este trabalho propõe uma forma de aprimoramento da usabilidade dos aplicativos embarcados em aparelhos celulares, formulando um novo método de interação com o usuário. O *método de interação Mandos* foi baseado na idéia de tarefas, que consiste em uma operação de uma função de um aplicativo a ser executada pelo usuário, e nas probabilidades de transição entre tarefas. As probabilidades de transição entre tarefas serão utilizadas para construção da interface com o usuário, e tais probabilidades são alteradas no decorrer do uso do aplicativo, fazendo com que a interação seja dinâmica. Para validar o método proposto foi desenvolvido um *framework*, chamado de *Mandos*, que adiciona algumas funcionalidades já prontas para o desenvolvedor de aplicativos que utiliza a tecnologia Java. Será apresentado um estudo de caso onde alguns protótipos de aplicativos embarcados em telefone móveis foram desenvolvidos, usando o *framework Mandos*, para a avaliação do método de interação. A avaliação foi baseada de modo a identificar qualquer relação de ganho ou perda de usabilidade dos aplicativos, tendo com avaliador os próprios usuários dos aplicativos embarcados em telefones móveis.

Palavras-chave: usabilidade, método de interação com o usuário. *framework* de desenvolvimento, desenvolvimento de aplicativos J2ME, telefonia móvel.

# Abstract

**Title:** Mandos: An Interaction Method with the User in Embedded Application in the Mobile Telephony

With the intense use of applicative in mobile device, the question “usability” begins to invigorate strongly as a study object, it is being considered a determinant factor of the success of this segment of the mobile computation. This work proposes an improvement way in the usability of the embedded applicative in mobile devices, considering a new method of user interaction. The *Mandos interaction method* is based on task idea, which consists in a possible operation that could be executed in an applicative by the user and probabilities between task changes. The cited probabilities will be used for construction of the user interface to interact dynamically with the user. To validate the considered method was developed a framework, called *Mandos*, which propitiates functionalities to the applicative developer that uses Java technology. A case study will be presented where some prototypes of embedded applicative mobile was developed, using *Mandos framework* for the interaction method evaluation. The evaluation was established in order to identify any relation of usability gain or loss of applicatives, having like approvers the own users of the embedded applicative in mobile telephones.

**Keywords:** usability, interaction method with the user, development framework, development of applicative J2ME, mobile telephony

# Lista de Figuras

Figura 2.1: As três dimensões que representam a experiência do usuário extraído de (Nielson, 1993).....	9
Figura 2.2: Curva de aprendizagem para usuário novato e experiente, extraído de (Nielson, 1993).....	9
Figura 3.1: Etapas do Método de Medida de Desempenho. Fonte: (Coopers, 1995) ..	19
Figura 3.2: Telefone móvel BenQ-Siemens SL75.....	24
Figura 3.3: Calculadora (aplicativo embarcado no telefone SL75) .....	25
Figura 3.4: Despertador (aplicativo embarcado no telefone SL75).....	26
Figura 4.1: Calculadora .....	33
Figura 4.2: Opções da calculadora .....	33
Figura 4.3: Calculadora utilizando o Método de Interação Mandos.....	34
Figura 5.1: Camada MIDP e CLDC, baseado em UFRGS (2006). .....	41
Figura 5.2: Camada do Framework Mandos entre as Camadas MIDP e CLDC, baseado em UFRGS (2006) .....	42
Figura 5.3: Ciclo de vida de um MIDlet .....	43
Figura 5.4: Hierarquia entre os componentes da arquitetura J2ME, baseada de (Martins 2002).....	44
Figura 5.5: Calculadora no modo expandido (aplicativo embarcado no celular BenQ-Siemens SL75) .....	45
Figura 5.6: Calculadora, quando Command Opções selecionado.....	46
Figura 5.7: Calculadora, quando em idioma inglês e alemão, respectivamente .....	49
Figura 5.8: Lista com alguns Commands com checkbox (“Alarme ligado” e “Pausa atom,”) indicando que a tarefa está no estado ativo. ....	51
Figura 7.1: Diagrama de Caixa e de Diamante dos valores do quesito Intuitividade.	63
Figura 7.2: Diagrama de Caixa e de Diamante dos valores do quesito Eficiência .....	65
Figura 7.3: Diagrama de Caixa e de Diamante dos valores do quesito Memorização.	67
Figura 7.4: Diagrama de Caixa e de Diamante dos valores do quesito Redução de Erros.....	70
Figura 7.5: Diagrama de Caixa e de Diamante dos valores do quesito Satisfação.....	72

# Lista de Tabelas

Tabela 2.1: Diferenças entre um aplicativo embarcado em um telefone móvel e um aplicativo instalado em um computador de mesa. Fonte: Adaptado de (Betiol, 2004), (Weiss, 2002) e (Chincholle, 2002).....	13
Tabela 4.1: Matriz de probabilidades de transição entre tarefas.....	13
Tabela 6.1: Distribuição dos participantes por faixa etária e sexo.....	55
Tabela 7.1: Notas atribuídas ao quesito Intuitividade para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos .....	62
Tabela 7.2: Dados estatísticos obtidos para o quesito Intuitividade .....	62
Tabela 7.3: Notas atribuídas ao quesito Eficiência para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos .....	64
Tabela 7.4: Dados estatísticos obtidos para o quesito Eficiência.....	65
Tabela 7.5: Notas atribuídas ao quesito Memorização para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos .....	66
Tabela 7.6: Dados estatísticos obtidos para o quesito Memorização .....	67
Tabela 7.7: Notas atribuídas ao quesito Redução de Erros para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos .....	68
Tabela 7.8: Dados estatísticos obtidos para o quesito Redução de Erros.....	69
Tabela 7.9: Notas atribuídas ao quesito Satisfação para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos .....	70
Tabela 7.10: Dados estatísticos obtidos para o quesito Satisfação .....	71

# Sumário

<b>1 Introdução .....</b>	<b>1</b>
1.1 Contexto .....	1
1.2 Definição do Problema.....	2
1.3 Objetivo.....	2
1.4 Motivação .....	2
1.5 O método de interação Mandos .....	3
1.6 Estrutura do trabalho .....	4
<b>2 Conceitos e Técnicas de Usabilidade .....</b>	<b>5</b>
2.1 Conceitualizando Usabilidade .....	5
2.2 Avaliando a usabilidade .....	6
2.2.1 Intuitividade.....	7
2.2.2 Eficiência.....	7
2.2.3 Memorização .....	7
2.2.4 Redução de Erros .....	7
2.2.5 Satisfação .....	8
2.2.6 Categorias de usuários e tipos de interfaces.....	8
2.2.7 Teste de Comparação .....	11
2.3 Influência da experiência precedente do usuário .....	11
2.4 Contexto dos aparelhos de telefone móveis .....	12
2.5 Trabalhos correlatos.....	13
<b>3 Métodos de Teste de Usabilidade em Aplicações para Telefones Móveis .....</b>	<b>17</b>
3.1 Método Pensamento em voz alta .....	17
3.2 Etapas do Método de Medida de Desempenho.....	18
3.3 Plano de Teste .....	19
3.4 O Avaliador .....	20
3.5 Selecionando os Participantes .....	20
3.6 Determinando o Número de Participantes .....	20
3.7 Material Utilizado Durante a Sessão de Teste .....	21
3.7.1 Roteiro do Avaliador .....	21
3.7.2 Questionário para Identificação do Perfil do Participante.....	22
3.7.3 Script de Orientação.....	22
3.7.4 Lista de Tarefas.....	22
3.7.5 Instrumentos para Coleta de Dados .....	23
3.7.6 Questionário de Avaliação do Sistema pelo Participante.....	23

3.7.7 Tópicos para Questionamento .....	23
3.7.8 Equipamente utilizado para os testes .....	24
<b>4 Método de interação Mandos .....</b>	<b>27</b>
4.1 Conceitos Básicos.....	27
4.1.1 Tarefa.....	27
4.1.2 Lista de Probabilidades .....	28
4.1.3 Matriz de probabilidades de transição entre tarefas .....	28
4.1.2.1 Matriz Estática.....	29
4.1.2.2 Matriz Dinâmica.....	29
4.1.2.2.1 Freqüência Relativa .....	30
4.1.4 Estudo do tempo de aprendizagem do usuário .....	30
4.1.5 Exemplo de uma matriz de probabilidade transição entre tarefas .....	31
4.2 Níveis de uso da Matriz de probabilidades de transição entre tarefas.....	32
4.3 Ilustrando o uso do Método de interação Mandos.....	32
4.4 Hipóteses a serem avaliadas sobre o Método de interação Mandos .....	35
<b>5 Framework Mandos .....</b>	<b>37</b>
5.1 Frameworks de Desenvolvimento de Software .....	37
5.2 Trabalhos correlatos em desenvolvimento de framework .....	39
5.3 Arquitetura J2ME (Java 2 Micro Edition) .....	39
5.3.1 Definição da Arquitetura J2ME .....	39
5.3.2 Breve visão geral da Arquitetura J2ME .....	42
5.3.3 Classe Command da Arquitetura J2ME.....	44
5.4 Armazenamento da matriz de probabilidade de transição entre tarefas.....	47
5.5 Utilizando o framework Mandos .....	52
5.5.1 Métodos do framework Mandos para tratar os Commands.....	53
<b>6 Descrição do Experimento.....</b>	<b>54</b>
6.1 Definição do produto a ser avaliado.....	54
6.2 Definição do contexto de uso .....	54
6.3 Definição dos objetivos e contextos de avaliação .....	54
6.3.1 Definição da amostra de participantes.....	55
6.3.2 Definição da lista das tarefas.....	56
6.3.3 Escolha das medidas e métricas de usabilidade .....	56
6.4 Condução dos ensaios de interação .....	57
6.4.1 Introdução do participante ao ambiente do teste .....	57
6.4.2 Treinamento do participante.....	58
6.4.3 Execução das tarefas .....	58
6.4.4 Questionários e discussões .....	58
6.5 Análise estatística .....	59
<b>7 Resultados .....</b>	<b>61</b>
7.1 Resultados do quesito Intuitividade.....	61
7.2 Resultados do quesito Eficiência .....	63
7.3 Resultado do quesito Memorização .....	65
7.4 Resultados do quesito Redução de Erros .....	67
7.5 Resultados do quesito Satisfação.....	69
7.6 Resultados gerais do Método de interação Mandos.....	72

7.7 Limitações do Experimento.....	73
7.7.1 Efeito de expectativa do sujeito – efeitos placebo e Hawthorne.....	74
7.7.2 Efeito de expectativa do experimentador .....	74
<b>8 Conclusão .....</b>	<b>75</b>
<b>Referências Bibliográficas .....</b>	<b>77</b>
<b>Anexo A: Plano de Teste.....</b>	<b>82</b>
<b>Anexo B: Roteiro do Avaliador .....</b>	<b>85</b>
<b>Anexo C: Questionário para Identificação do Perfil do Participante do     Teste de Usabilidade .....</b>	<b>88</b>
<b>Anexo D: Script de Orientação.....</b>	<b>90</b>
<b>Anexo E: Lista de Tarefas .....</b>	<b>92</b>
<b>Anexo F: Coleta de Dados pelo Avaliador.....</b>	<b>94</b>
<b>Anexo G: Questionário de Avaliação do Participante .....</b>	<b>98</b>
<b>Anexo H: Plano de Teste.....</b>	<b>101</b>

# Capítulo 1

## Introdução

Este capítulo apresenta uma iniciação a este trabalho buscando contextualizar, definir o problema abordado, apresentar os objetivos, a motivação deste trabalho e como o mesmo está estruturado.

### 1.1 Contexto

O avanço da tecnologia e o aumento da capacidade de processamento dos componentes eletrônicos vêm permitindo o surgimento de novas concepções do telefone móvel, criando a possibilidade de o mesmo atuar como um computador de mão, que pode carregar e executar aplicativos embarcados a serem utilizados pelos usuários da tecnologia telefonia móvel, que é considerada uma das tecnologias de comunicação que mais cresceram em todos os tempos.

A interação homem-máquina não se limita à interação do usuário com as páginas da internet e dos aplicativos para o computador de mesa. Segundo Ipiña (2002), o computador encontra-se embarcado e invisível em diversos produtos utilizados diariamente, tornando-se onipresente, como, por exemplo, já ressaltado na telefonia móvel. Assim sendo é necessário estudar e cada vez mais aprimorar o meio desta constante interação.

Segundo Taurion (2003), a aceitação do serviço tende a ser o elo mais lento desse processo. Além da utilidade, disponibilidade e custo, um dos principais elementos para a aceitação do usuário refere-se à usabilidade das interfaces e equipamentos.

Segundo Chittaro (2001), a usabilidade dos equipamentos e aplicações para a telefonia móvel é um fator determinante do sucesso deste segmento da computação móvel.

Desta forma, Betiol (2004) conclui que a usabilidade deve ser considerada como um dos principais requisitos de projeto para que as aplicações e serviços atendam às necessidades do usuário móvel, permitindo que ele seja eficaz, produtivo no uso do tempo e dos recursos e esteja satisfeito em relação aos atributos do sistema.

## 1.2 Definição do problema

Segundo Kim (2005), o modo de utilizar um aparelho celular varia de acordo com a cultura, faixa etária, localização geográfica, características intrínsecas do usuário, além de diversos outros fatores, ou seja, há uma variação das funções mais utilizadas de um aplicativo, assim como variações nas opções mais utilizadas. Isto faz com que seja difícil desenvolver um aplicativo embarcado que tenha uma boa usabilidade em termos globais.

Uma característica no modo atual de desenvolvimento de aplicativos para telefonia celular é que a interface de interação com o usuário está totalmente definida no código, ou seja, para que haja alteração no modo de exibição de um dado aplicativo é necessário reescrever trechos do código fonte. Sendo assim, há a necessidade de ter diferentes versões de um mesmo aplicativo para agradar distintos públicos.

Outro problema inerente da forma atual de codificação é que atualmente os menus dos aplicativos embarcados são estáticos, ou seja, não se modificam ou se reestruturam de forma a se adaptar ao uso do usuário.

A pergunta a ser respondida é se a atual forma de interação com o usuário é a maneira mais eficiente. E se ao propor uma nova maneira de interação com o usuário, que além de resolver o problema de várias versões de um código e não adaptação do aplicativo ao modo de utilização de um povo e de um usuário em particular apresenta uma forma mais eficaz de executar as tarefas mais comuns de um aplicativo, o mesmo método será aceito com maior satisfação que o modo normal.

## 1.3 Objetivo

O objetivo principal deste trabalho é apresentar uma solução em que a usabilidade de um aplicativo possa ser aprimorada, alterando um arquivo externo do programa, isto é, evitando a modificação no código fonte de um aplicativo, assim, com uma única versão de um aplicativo várias formas de apresentação da interface do usuário poderão ser geradas de acordo com o corrente público-alvo.

Além de ser uma forma de evitar a geração de várias versões de aplicativos, tem também por intenção modificar a forma de interação com o usuário, ou seja, após o uso de uma ação de um aplicativo, a próxima contribuição a ser apresentada será a de maior probabilidade de o usuário executá-la. Com isto, será pressionado um número menor de botões para atingir a funcionalidade desejada. Assim pretende-se que a usabilidade seja aprimorada com este novo método de interação.

## 1.4 Motivação

A primeira geração de dispositivos móveis foi desenvolvida para satisfazer a mesma necessidade do usuário das linhas de telefones fixos, isto é, fazer e receber chamadas. No entanto, muitas funções têm sido adicionadas, tais como troca de mensagens texto/voz/multimídia, conexão Internet, tocador de música, filmadora, câmera, cartão de crédito, agenda, jornal, *bluetooth*; se estendendo as funcionalidades nunca antes imaginadas a esses pequenos dispositivos que estão evoluindo na direção dos telefones inteligentes (Ferreira, 2006).

De forma a tornar o telefone e todas suas funcionalidades efetivamente usáveis, o usuário precisa encontrar um dispositivo recheado com aplicativo que apresentem uma boa usabilidade. Torna-se necessário entender as implicações de *design* e usabilidade do mundo móvel que se iniciou há mais de uma década.

Primeiramente, as limitações dos dispositivos móveis foram identificadas como limitações tecnológicas, tais como recursos computacionais pobres, se comparados com outros recursos computacionais estáticos, limitada fonte de energia, pouca confiabilidade de conexão de rede e pouca memória. Mais tarde, vários aspectos de interação humana com computadores móveis foram considerados, tais como: segurança, limitações ergonômicas, propriedades de acesso ubíquo e colaboração em ambientes móveis. Nos últimos anos, a discussão tem se voltado para como a usabilidade tem acompanhado o avanço tecnológico nesses pequenos dispositivos que apresentam cada vez mais uma maior complexidade.

O acesso a qualquer hora, qualquer lugar e de qualquer maneira para qualquer um apresenta mais desafios do que originalmente se havia pensado. Enquanto muitas restrições tecnológicas estão somente há poucos passos de serem solucionadas, um enorme número de restrições ambientais (interação e interface) e algumas limitações humanas ainda precisam ser resolvidos.

O principal objetivo desta dissertação é apresentar um novo método de interação com o usuário, o *método de interação Mandos*, e validar a sua eficiência em relação à melhoria da usabilidade de um aplicativo.

## 1.5 O método de interação Mandos

O *método de interação Mandos* pressupõe que um dado aplicativo é composto de *tarefas* a serem executadas pelo usuário e para a execução de cada *tarefa* há um determinado meio do usuário acioná-la, pode-se inferir que após a execução de uma dada *tarefa* há uma determinada probabilidade de execução de outra *tarefa* ou a até a mesma. Baseando-se nisto a proposta é a criação de uma ferramenta que auxilie na construção de UI (*user interface*, interface com o usuário) através da elaboração de uma matriz que conterà dados relativos à probabilidade de transição entre as *tarefas* de um aplicativo e, com base nesta matriz, o citado meio de ativação de uma dada *tarefa* será construído e disposto de maneira que o usuário despenda o menor esforço para a sua ativação.

Para que haja o correto uso da matriz proposta é necessário que o aplicativo esteja projetado para o uso da mesma, ou seja, um padrão de desenvolvimento deve ser conjuntamente proposto.

Além disso, para validar o novo método de interação com o usuário, propõe-se a criação de um *framework* de desenvolvimento focado na usabilidade, que poderá ser utilizado em três níveis de complexidade de utilização do *framework*:

- a) O primeiro nível de uso consiste na consulta de um estado inicial que será utilizado para a criação da interface com o usuário, ou seja, no início de um aplicativo, dados de um arquivo externo serão coletados, a idéia é que os citados dados externos poderão ser facilmente modificados, sendo assim a interface também será modificada, estabelecendo-se assim várias formas de apresentação sem que haja alteração no código;

- b) O segundo nível apresentaria uma constante alteração da interface do usuário, de acordo com o corrente estado do aplicativo, neste nível uma nova maneira de interação com o usuário seria estabelecida;
- c) O terceiro nível seria um aprimoramento do segundo, ou seja, os dados de transição entre os estados seriam alterados de acordo com o uso do aplicativo, de maneira a se adaptar a um específico usuário.

Constituindo assim três modos de interação com o usuário de acordo com o nível de utilização do *framework*. Neste trabalho também iremos classificar os modos de interação em três níveis, conforme classificado anteriormente para o *framework Mandos*. Sendo que no primeiro nível, o conceito de interação será praticamente inalterado para o usuário, pois os dados serão carregados no início e serão mantidos estático. No segundo e terceiro nível haverá já uma mudança perceptível ao usuário da aplicação. Neste trabalho o foco será validar o *método de interação Mandos* nos níveis 2 e 3, devido ao fato citado de propor uma perceptível e nova maneira de interação com o usuário.

## 1.6 Estrutura do trabalho

Este trabalho está organizado em 8 capítulos.

No capítulo 2, Conceitos e Técnicas de Usabilidade, procura-se realizar uma definição do tema usabilidade e apresentar o contexto da telefonia móvel além de fazer um breve resumo de todos os atuais esforços na área de melhoria e avaliação da usabilidade para telefones móveis, procura-se sempre realizar um paralelo entre os objetivos deste corrente trabalho e do trabalho analisado já publicado, deixando clara a contribuição dada pelo trabalho.

No capítulo 3, Método de Teste de Usabilidade em Aplicações para Telefone Móvel, faz-se uma explanação sobre os métodos utilizados para teste de usabilidade, focando nos testes voltados para aplicativos embarcados em telefones móveis. Procura-se neste capítulo definir e apresentar a parte teórica necessária para realização e comprovação do experimento conduzido neste trabalho.

No capítulo 4, Método de interação Mandos, apresenta-se os conceitos básicos para o entendimento do método proposto, assim como exemplifica e define o citado método de interação com o usuário.

No capítulo 5, Framework Mandos, faz-se uma breve conceituação de *framework*, enfocando o tipo e funcionalidade a ser desempenhada pelo *framework* proposto, apresenta-se também a arquitetura J2ME (Java 2 *Micro Edition*), de forma bem compactada, arquitetura esta sobre a qual o *framework* proposto será definido e apresenta-se o *framework* proposto, ressaltado sua funcionalidade e objetivos.

No capítulo 6, Descrição do Experimento, descreve o experimento realizado para investigar qual a influência exercida pelo uso do Método de interação Mandos na usabilidade de um aplicativo embarcado em um telefone móvel.

No capítulo 7, Resultados, apresentam-se os resultados obtidos do experimento realizado.

No capítulo 8, Conclusão, onde são apresentadas as conclusões relativas a este trabalho. Este capítulo também sugere trabalhos futuros.

# Capítulo 2

## Conceitos e Técnicas de Usabilidade

O intuito deste capítulo é apresentar os principais conceitos para a usabilidade e citar de forma concisa os atuais esforços realizados atualmente nesta área, realizando um paralelo com o objetivo deste trabalho.

### 2.1 Conceitualizando Usabilidade

Apesar de um software ou hardware apresentar uma funcionalidade correta e útil, se o mesmo não apresentar uma maneira consistente e adequada de interação com o usuário, estará fadado a não aceitação do produto pelo usuário. Em suma, a maneira do usuário interagir com um software ou hardware é um fator decisivo no seu sucesso.

É necessário, então, definir de forma clara o que vem a ser a usabilidade. Usabilidade é um conceito que está implicitamente compreendido pela maioria das pessoas. Raramente nota-se o esforço despendido para alcançar um produto bem projetado, mas as falhas de projeto são rapidamente notadas, o usuário é bem mais crítico quando o aplicativo apresenta alguma falha, condenando ao desuso um aplicativo. Porém, usabilidade é um conceito difícil de definir adequadamente, porque muitos aspectos contribuem para a sua percepção. Para tornar mais clara a usabilidade apresenta-se a seguir três definições sobre a mesma.

Shackel (1991) concebeu um modelo de percepção de produtos baseado na aceitação. Esta aceitação será diretamente dependente da utilidade, da usabilidade, da capacidade de agradar e dos custos, onde:

- **Utilidade:** concerne ao mapeamento das necessidades dos usuários com as funcionalidades do produto;
- **Usabilidade:** diz respeito à facilidade de utilização prática das funcionalidades do produto;
- **Capacidade de agradar:** avaliações subjetivas do usuário;
- **Custos:** tanto encargos econômicos quanto conseqüências sociais e organizacionais decorrentes da aceitação do produto;

Nielsen (2003) define usabilidade através de cinco aspectos principais, aprendizado, eficiência, memorização, redução de erros e satisfação, onde:

- **Aprendizado:** aprender a utilizar um novo sistema deve ser fácil suficiente para que o usuário comece a usá-lo rapidamente;
- **Eficiência:** o sistema deve ser eficiente de tal forma que os usuários possam realizar muitas tarefas rapidamente;
- **Memorização:** um usuário ocasional do sistema não precisa reaprender a utilizar o sistema, isto é, o sistema precisa ser de fácil memorização;
- **Redução de Erros:** o sistema deve ter capacidade de recuperação de erros e deve parar o usuário quando este cometer erros;
- **Satisfação:** o usuário deve ficar satisfeito com o uso do sistema.

A ISO 9241 (1998), *Ergonomic Requirements for Office Work with Visual Display Terminals*, é um padrão internacional que define usabilidade como a efetividade, eficiência e satisfação com que o usuário atinge metas específicas em ambientes específicos, onde:

- **Efetividade:** diz respeito à precisão e à completude com a qual os usuários alcançam metas específicas;
- **Eficiência:** diz respeito aos recursos necessários em face à completude e precisão para que os usuários alcancem suas metas;
- **Satisfação:** diz respeito ao conforto e à aceitação do uso do sistema pelos usuários.

Pela definição de Shackel, o *Método de interação Mandos* estará auxiliando na usabilidade, pois proporcionará uma nova forma de interação, onde as tarefas mais executadas tendem a tornar-se mais fácil de utilizar.

Pela ISO 9241, atua-se na *eficiência*, proporcionando ao usuário despende o mínimo de recurso possível para realizar a tarefa mais comum de um aplicativo, a *satisfação* deve ser ainda avaliada.

Após ser apresentado as três definições mais difundidas sobre usabilidade, optou-se neste trabalho, por utilizar a definição de Nielsen para posterior análise dos resultados obtidos, por oferecer uma maior gama de aspectos a serem analisados e com isto pode explicitar o possível sucesso ou falha do *framework* proposto, devido a isto a citada definição será tratada mais detalhadamente na próxima subseção.

## 2.2 Avaliando a usabilidade

A forma mais comum de se avaliar a usabilidade de um software é observando a sua interação com o usuário, podendo essa observação ser feita em laboratório, com uma quantidade representativa de usuários para o qual o sistema foi desenvolvido, ou no próprio ambiente de trabalho onde o sistema será implantado. O mais importante nesse processo avaliativo é que, sempre que possível, deve-se utilizar o usuário certo para as tarefas certas, a fim de se obter o máximo de desempenho avaliativo.

Nos casos em que o software alcance uma variedade de usuários com características diferentes, por exemplo, os softwares de prateleira, deve-se elaborar uma avaliação que utilize escalas de valores para cada critério analisado, observando, de acordo com Nielsen (1993), “.....em uma escala de 1 a 5”.

Os itens que definem a usabilidade deverão ser identificados e explicados aos usuários que irão testar o aplicativo. Como será utilizado o conceito de Nielsen para usabilidades, os itens a serem medidos são: intuitividade, eficiência, memorização, redução de erros e satisfação.

### 2.2.1 Intuitividade

Entre os atributos que compõe a usabilidade, o mais importante é a intuitividade. Para que ocorra minimamente uma interação, a interface deve apresentar características que facilitem sua utilização permitindo que usuários básicos ou avançados possam aprender seus recursos de IHC (interação homem-computador) de forma clara e objetiva.

O sistema deve apresentar facilidade de uso permitindo que, mesmo um usuário sem experiência, seja capaz de produzir algum trabalho satisfatoriamente.

Sua avaliação não apresenta muita complexidade, basta utilizar uma amostragem de usuários, pertencentes ao grupo a que se destina o software e que não tenha utilizado o sistema, medindo-se o tempo necessário para que o mesmo tenha um bom controle em sua utilização. Essa medição tem como objetivo informar a eficiência do nível de interação, quanto menor for o tempo gasto, melhor será a qualidade da apreensibilidade.

Neste trabalho o tempo não será medido com valores, mas será indicado ao usuário e o mesmo deverá, ao avaliar a intuitividade, indicar qual funcionalidade dependeu mais tempo para aprender.

### 2.2.2 Eficiência

A eficiência é determinada através da medição do tempo gasto na realização de tarefas de um aplicativo por usuários experientes.

Como a avaliação a ser realizada será uma comparação entre métodos de interação, o valor tempo não será registrado, cabendo ao avaliado indicar qual método ele obteve melhor desempenho, ou seja, executou uma tarefa em menor tempo.

### 2.2.3 Memorização

A memorização pode ser medida através do registro do tempo decorrido desde a última interação e do tempo gasto para executar uma tarefa específica. Outra forma de medi-la é verificar se, após um determinado tempo de interação, o usuário é capaz de reconhecer comandos e ações específicas através de questionário. Como a avaliação ser realizada será uma comparação entre aplicativos, o valor não será registrado, ou seja, caberá ao usuário do aplicativo mensurar isto intuitivamente, indicando em que método o mesmo apresentou um melhor desempenho em relação a memorização.

### 2.2.4 Redução de Erros

É considerado erro qualquer ação que leve o usuário a não executar determinada tarefa não se levando em conta os possíveis impactos causados por diferentes tipos de erros.

A quantidade de erros apresentados pelo sistema deve ser o mais reduzido possível, além disso, eles devem apresentar soluções simples e rápidas mesmo para usuários iniciantes. Erros graves ou sem solução não podem ocorrer.

Alguns erros são considerados catastróficos quando não podem ser detectados pelo usuário, resultando em uma produção defeituosa ou quando causam perdas irreversíveis de informações. Esse tipo de erro deve ser tratado separadamente e com um maior grau de atenção.

### 2.2.5 Satisfação

A satisfação representa o quão agradável deve ser a interação do usuário com o sistema. Por ser uma característica subjetiva da usabilidade, ela é medida através da aplicação de questionários individuais, devendo ser levado em consideração a média das respostas obtidas de um determinado grupo de usuários.

Essa característica deve ter especial importância para aplicativos do tipo educacionais, jogos, interação e aplicações domésticas, pois, necessitam de interfaces mais atrativas e/ou divertidas. Em algumas dessas aplicações o item diversão é mais importante que a velocidade de resposta às solicitações requeridas, desde que o usuário tenha tempo disponível para divertir-se (Carroll, 1988).

É importante observar que o atributo satisfação é bastante subjetivo e diferente dos demais critérios de usabilidade, pois cada usuário apresenta características individuais que podem variar de atitudes diante do computador (Nielsen, 1988).

Segundo Nielsen (1993), em princípio algumas medições objetivas eram realizadas com o intuito de detectar o nível de satisfação em relação à interface, dentre essas, pode-se destacar a utilização de EEG (Eletroencefalograma), dilatação de pupila, batimentos cardíacos, pressão sanguínea e nível de adrenalina no sangue. Contudo, tais experimentos alteravam o estado normal de cada usuário analisado, causando constrangimento e intimidação ao serem conectados a eletrodos ou tomando-lhes amostra de sangue, ações estas que acabavam por interferir nos resultados obtidos.

Como alternativa, a satisfação subjetiva do usuário pode ser mensurada utilizando a observação e a aplicação de questionários. Embora as respostas individuais sejam subjetivas, quando realizadas com diversos usuários pode-se obter medidas objetivas de satisfação em relação ao sistema.

### 2.2.6 Categorias de usuários e tipos de interfaces

Outros fatores de grande importância que devem ser considerados são as categorias de usuários e suas diferenças individuais. A Figura 2.1 mostra, segundo Nielsen (1993), as três principais dimensões que distinguem o usuário em sua perícia: com o sistema, com computadores em geral e com o domínio da tarefa.

Quando se discute perícia do usuário, deve-se levar em consideração a experiência deste em relação a uma interface específica e, normalmente, são classificados em novatos ou experientes ou se localizam em algum lugar entre ambos.

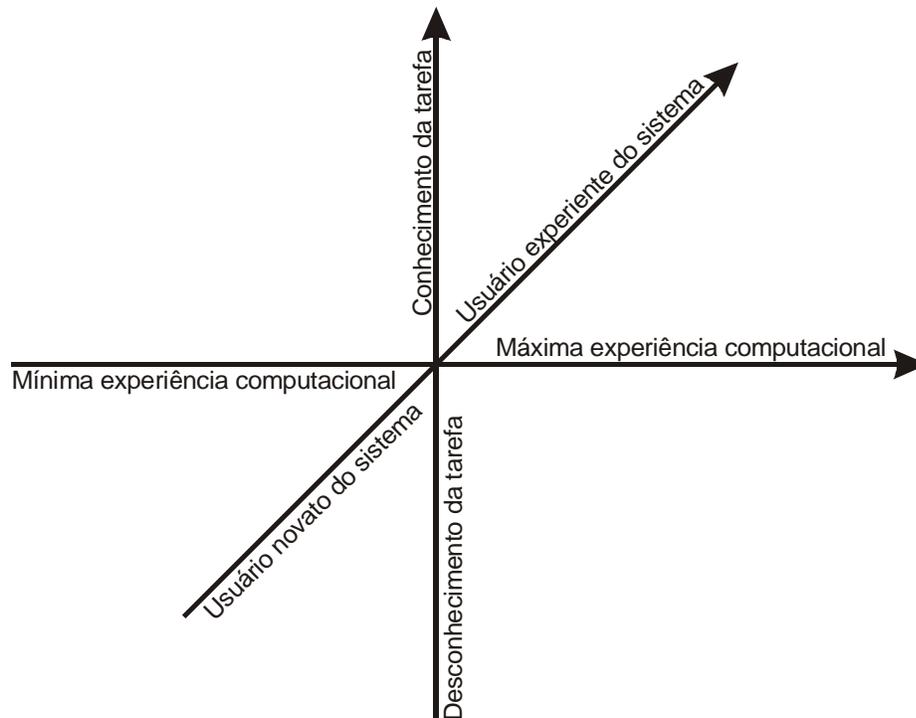


Figura 2.1: As três dimensões que representam a experiência do usuário extraído de (Nielson, 1993)

A curva de aprendizagem que ocorre tanto com o usuário novato como com o experiente ou perito pode ser representada pela Figura 2.2.

Considerando um sistema hipotético, um usuário novato deve apresentar fácil aprendizagem e pouca eficiência na utilização. Para o caso de usuários experientes o sistema pode apresentar maior dificuldade de aprendizagem e alta eficiência na utilização.

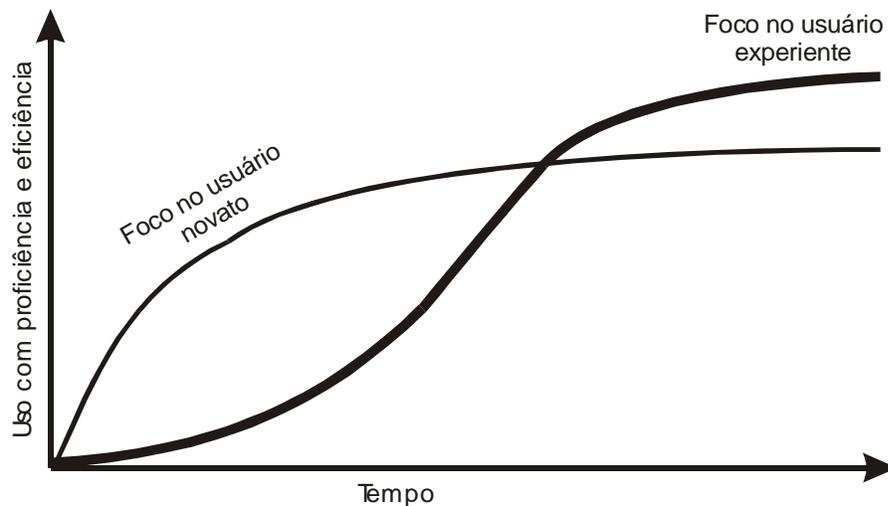


Figura 2.2: Curva de aprendizagem para usuário novato e experiente, extraído de (Nielson, 1993)

Há vários elementos de interface que podem colaborar com o aumento da perícia do usuário sem interferir nas ações pretendidas de usuários novatos:

- Teclas de atalho – opções que são indicadas nos menus, na forma de combinação de duas ou mais teclas que apresentam a mesma função desejada.
- Barra de ferramentas – conjunto de ícones que executam a mesma tarefa encontrada em diversas opções nos menus.
- Macros – utilização de um conjunto de comandos com o objetivo de realizar uma tarefa com um determinado nível de complexidade.
- Sistemas de ajuda On-line – permite a qualquer nível de usuário aprofundar seus conhecimentos nas técnicas de interação disponíveis em um determinado ambiente, sugerindo melhores alternativas e caminhos para alcançar um objetivo. Sistemas de ajuda colaboram com a transição do usuário novato para experiente.

Existem interfaces desenvolvidas com o objetivo de proporcionar a interação com usuários novatos em situações onde, quase sempre, serão utilizadas apenas uma vez, são os casos de quiosques que possuem informativos multimídias e estão dispostos em lugares públicos com dados gerais de localização, entre outros.

Outras necessitam de uma série de leituras de instruções passo a passo, basicamente apresentando o mesmo formato, que conduzem o usuário na execução de uma tarefa objetiva, tais como programas de instalação, rotinas de formatação de discos, cópias de arquivos, etc.

Alguns sistemas apresentam interfaces com dois grupos distintos de menus. Os menus curtos, desenvolvidos para usuários novatos, que apresentam funções básicas, porém com recursos limitados e os menus longos criados para os usuários experientes, apresentando uma grande variedade de recursos. Este modelo permite que a interface disponibilize uma vasta gama de recursos aos usuários experientes sem interferir na interação do novato. A maioria das interfaces, contudo, são desenvolvidas tanto para novatos como para experientes e apresentam a característica de possuir os dois estilos.

É importante destacar que, em sistemas que apresentam certo grau de complexidade, nenhum usuário apresenta perícia completa em um nível de abrangência total, ou seja, até mesmo um usuário experiente deve apresentar características de usuário novato ao interagir com partes específicas do sistema aos quais não esteja familiarizado. Desta forma, mesmo usuários experientes podem necessitar de auxílio na compreensão de tarefas desconhecidas e interfaces não utilizadas comumente.

Outro fator que influencia no desenvolvimento de interfaces é o nível de domínio da tarefa por parte do usuário na utilização de um determinado sistema. Se o domínio do usuário for extenso pode-se usar terminologias específicas e uma alta densidade de informação na tela. Nos casos em que o conhecimento do domínio da tarefa é diminuto se faz necessário o desenvolvimento de interfaces mais explicativas com terminologias menos densas (Nielsen, 1993).

Deve-se levar em consideração, também, o fator de aceitação/rejeição da informática por parte dos usuários. Existem usuários que dedicam uma grande parcela de tempo no aprendizado total de seu sistema buscando o domínio amplo do conhecimento em todos os seus níveis. Em posição contrária, alguns usuários são extremamente avessos ao computador e seus recursos, podendo gerar processos de rejeição em diversos níveis de intensidade. Para estes usuários, a interface deve ser

projetada para apresentar um ambiente o mais amigável possível objetivando minimizar ou reverter tais processos.

Observa-se, partindo do exposto acima, que a usabilidade aplicada à interface depende de diversos fatores de ordem subjetiva, tais como: nível de qualificação do usuário, tipo de aplicação, capacidade de aprendizagem, aceitação/rejeição, domínio da tarefa, complexidade do sistema, entre outros. Além disso, em alguns casos, a interface é gerada a partir de suposições empíricas do projetista ou baseada na opinião do usuário, o qual pode não apresentar qualificação para tal. Essa prática, na maioria das vezes, contempla poucos ou nenhum dos critérios acima descritos, ocasionando uma baixa qualidade na interface.

Tanto a usabilidade como seus critérios devem ser estudados de forma mais sistemática, permitindo que seja feito um projeto adequado com todas as suas etapas bem estabelecidas.

### **2.2.7 Teste de Comparação**

O teste de usabilidade a ser utilizado é do tipo comparação, pois serão comparados os aplicativos com o uso do método de interação Mandos e sem o método Mandos, este tipo de teste não é associado a nenhum ponto específico do ciclo de desenvolvimento de um produto.

Nos primeiros estágios pode ser usado para comparar diferenças entre estilos de interface através do teste de exploração; nos ciclos intermediários, pode ser usado para medir a efetividade de um elemento integrante da interface; no final do ciclo, um teste de comparação pode ser usado para ver como a liberação de um produto atinge um produto concorrente.

O teste de comparação pode ser usado como uma junção dos demais testes sendo que seu objetivo é realização de comparações em todos os níveis.

## **2.3 A influência da experiência precedente do usuário no desenvolvimento de um aplicativo**

Sacher (2002) tenta encontrar um novo paradigma para os serviços móveis. Para tanto, estudaram se um novo paradigma que está baseado no uso do computador pessoal, no uso do telefone, uma mistura de ambos ou algo completamente novo. Pesquisas realizadas usando as regras para os paradigmas do computador pessoal ou telefone apontaram que os citados paradigmas não podem ser aplicados ao contexto móvel.

Entretanto, Sacher (2002) começa um projeto de pesquisa focado em um serviço móvel para jovens. Concluíram que os serviços poderiam ser melhores aceitos se fossem várias formas de serem apresentados, de acordo com o uso e usuário.

O projeto de dispositivos que obtenham boa aceitação será baseado na descoberta do paradigma mais eficaz para o uso. O uso é provavelmente dependente de um grande número de fatores que não podem ser previstos sem pesquisa empírica usando simulações no mundo real destes aplicativos.

A obtenção de dados e utilização do aplicativo para se adaptar ao usuário é uma boa proposta para se obter uma aceitação do mesmo pelo usuário.

## 2.4 Contexto dos aparelhos de telefone móveis

O tamanho de tela dos telefones móveis é bastante pequeno comparado ao PDA comum e só recentemente telas coloridas de alta qualidade foram oferecidas ao público a um preço acessível.

O método de entrada de dados em um telefone é realizado pelo uso do teclado do telefone, podendo oferecer assim em torno de 22 teclas. Embora para navegação ao redor de um aplicativo no telefone móvel 6 teclas são usadas; uma tecla de seleção, uma de voltar e 4 teclas de navegação para cima, abaixo, esquerda e direita.

Haitani (2000) contrastou padrões de uso de dispositivos portáteis e dispositivos de laptop. Pessoas geralmente usam dispositivos portáteis para ter acesso a dados rapidamente, sendo seu tempo de utilização rápido; em contraste com laptops que tendem a armazenar uma quantidade de dados que são menos freqüentemente usados, mas para sessões de utilização mais longas (por exemplo, trabalhando em uma planilha eletrônica). Considerando que o padrão de uso é diferente, Haitani descreve que o projeto de interface com o usuário deveria ser diferente. Ele discute que a maneira para garantir a satisfação do usuário é reduzir a frustração de procurar por menus, entendendo as prioridades do usuário e oferecendo-as de forma mais atraente em termos da visibilidade e arquivos freqüentemente usados. Ele focaliza em aperfeiçoar a navegação para reduzir o número de teclas a serem pressionadas para acessar arquivos freqüentemente usados, assim como acessar funções mais executadas. Ele afirma também que os usuários não se importam em pressionar uma quantidade maior de botões para alcançar para funcionalidades menos freqüentes.

Além das características físicas dos telefones móveis, é muito importante entender o usuário móvel e o contexto em que ele está inserido (Betiol, 2004).

Enquanto o computador de mesa é usado para digitar texto, elaborar planilhas eletrônicas, fazer pesquisas na *Web*, ou seja, tarefas que exigem concentração e são executadas durante um longo período de tempo, os aplicativos embarcados em telefones móveis são voltados para aplicações mais rápidas. O tempo é um fator muito importante para o usuário móvel. Ele é mais impaciente e exigente que o usuário do computador de mesa e tende a utilizar serviços que permitam uma manipulação rápida da interface e um acesso à informação através de um número reduzido de passos (Ericsson, 2001).

O ambiente de uso de um aplicativo em um telefone móvel é normalmente muito mais dinâmico em relação ao ambiente de uso do computador de mesa e, portanto, demanda a atenção do usuário para outras tarefas além da interação com o equipamento. O usuário de um aplicativo embarcado em um telefone móvel normalmente está envolvido em várias atividades que ocorrem simultaneamente. A sua atenção está dividida entre o uso do equipamento, as outras atividades que ele está realizando e o ambiente que o cerca (Chincholle, 2002). Desta forma, o foco principal não é a interação com o equipamento, mas a busca da informação enquanto ele interage com o ambiente a sua volta. A Tabela 2.1 apresenta as principais diferenças entre um aplicativo embarcado em telefone móvel e um aplicativo em um computador de mesa.

Segundo Rischpater (2000) “o usuário móvel tem menor capacidade de processar e absorver conteúdo que um usuário que está sentado em frente a um computador de mesa”.

Para Chincholle (2002) a interação com o equipamento é uma tarefa secundária que não deve interferir com a tarefa principal que ele está realizando como caminhar.

Os aplicativos embarcados em telefones móveis estão destinados a atingir uma grande variedade de usuários, entre eles estão aqueles que nunca utilizaram o computador, ou que têm pouca experiência no seu uso. A “usabilidade imediata” é muito importante para esses usuários (Mackenzie, 2002). Neste contexto, pode não ser apropriado fazer uso de modelos e metáforas de interface com o usuário que são amplamente utilizadas no computador de mesa. Além de não estar familiarizado com esses elementos, o usuário pode considerar o equipamento muito complicado e não estar disposto a aprender formas complexas de interação (Mohageg, 2000).

Segundo Betiol (2004), as necessidades e características do usuário móvel, o contexto de uso dos computadores de mão, as características das aplicações e as limitações físicas desses equipamentos são fatores que irão influenciar a interação e devem ser considerados tanto no projeto das interfaces quanto na avaliação de usabilidade destes equipamentos.

Tabela 2.1: Diferenças entre um aplicativo embarcado em um telefone móvel e um aplicativo instalado em um computador de mesa. Fonte: Adaptado de (Betiol, 2004), (Weiss, 2002) e (Chincholle, 2002)

	<b>Aplicativo embarcado em um telefone móvel</b>	<b>Aplicativo instalado em um computador de mesa</b>
<b>Uso</b>	Consulta rápida à informação em várias situações diferentes, normalmente longe do computador de casa ou do escritório. Comunicação rápida por voz e/ou mensagem de texto.	Longas e complexas tarefas de processamento da informação. <i>Web browsing</i> e <i>e-mail</i> .
<b>Freqüência / Duração</b>	Várias vezes durante o dia com curta duração.	Poucas vezes durante o dia durante longo período de tempo.
<b>Mobilidade</b>	Pode ser facilmente operado com as mãos sem necessidade de apoio. Pode ser levado no bolso com facilidade.	Necessita de uma mesa. Precisa estar conectado por cabos ou tem necessidade de recarga freqüente. Alguns modelos são portáteis.
<b>Entrada de dados</b>	Diversas: Minitelclados, caneta, botões de navegação.	Padrão: Teclado completo, mouse, tela de toque.
<b>Display</b>	640X480 <i>pixels</i> para menos.	640X480 <i>pixels</i> para mais.
<b>Capacidade de Armazenamento</b>	Normalmente pouca. Alguns equipamentos possuem expansão de memória.	Praticamente ilimitada. Disquetes, CDRW, DVD RAM, disco rígido.
<b>Conexão</b>	Lenta e instável (por enquanto).	Veloz e estável.

## 2.5 Trabalhos correlatos em usabilidade

Apresenta-se a seguir uma compilação dos atuais estudos sobre usabilidade para dispositivos móveis, onde se busca descrever alguma analogia com o estudo proposto, procurando, sempre que possível, apresentar e enaltecer os principais pontos de divergência e convergência do foco de estudo.

Em Shen (2001) é possível encontrar uma revisão da literatura sobre a engenharia de requisitos para o desenvolvimento de sistemas para dispositivos móveis. Esse artigo concentra o esforço da usabilidade de dispositivos móveis para a fase de requisitos, no qual as definições de escopo, funcionalidades e elaboração de um aplicativo estão sendo realizadas, ou seja, para o início do ciclo de desenvolvimento do produto. Os autores selecionaram 07 estudos, a partir dos quais formularam algumas sugestões a serem seguidas durante a condução da fase de análise de requisitos, dentre as quais se destacam:

- i) Reunião de requisitos baseados no contexto;
- ii) Prototipação como meio de deixar claro o que os usuários querem e o que está sendo desenvolvido;
- iii) Realização de avaliações formais;
- iv) Análise de cenários;
- v) Delineamento do perfil dos usuários.

As conclusões a que chegam os autores deste artigo não são surpreendentes ou diferem substancialmente das considerações vigentes da engenharia de requisitos para o desenvolvimento de sistemas não móveis.

Nota-se que o delineamento do perfil do usuário foi um fator considerado na fase de requisito. Neste ponto é que o *método de interação Mandos*, proposto neste trabalho, tenta se adequar, propondo uma forma prática de criar várias versões de um aplicativo voltado para cada público alvo sem haja um grande esforço, ou seja, o uso do mesmo já poderia auxiliar a elaboração dos requisitos, podendo atingir de forma planejada a vários perfis simultaneamente.

Klockar (2003) conduziu um estudo de usabilidade de telefones celulares focalizado na facilidade de uso de funcionalidades típicas e pouco freqüentes. Os testes contaram com a participação de 9 usuários utilizando seus próprios celulares para executarem 26 tarefas de testes. Dentre os dispositivos testados estavam 2 modelos da *Nokia* e 2 modelos da *Siemens*.

Culturalmente, segundo os autores, é dito que os celulares da *Nokia* são mais fáceis de utilizar que os celulares da *Siemens*, no entanto, nenhuma diferença quanto à facilidade de uso dos dispositivos destas marcas foi encontrada com base no estudo realizado por Klockar (2003). Este estudo chegou a resultados tais como um histograma representando o número médio de botões a serem pressionados para a realização das tarefas e a conclusão de que, apesar dos fabricantes estarem preocupados com mecanismos de atalho ou de acesso rápido, os usuários raramente utilizam esta forma de acesso.

O estudo também concluiu que a usabilidade de telefones celulares pode ser aprimorada a partir do projeto cuidadoso dos menus e da navegação e da consideração do modelo mental do usuário no projeto do dispositivo.

Este artigo é de grande influência para sustentar a idéia proposta do *método de interação Mandos*, ou seja, a importância da navegação e elaboração dos menus, e é justamente isto que será proposto no *método de interação Mandos*, ou seja, uma forma de facilitar ao usuário o acesso às tarefas mais comuns.

Maybury (2001) define *interfaces inteligentes com o usuário* como entidades que tentam realçar a flexibilidade, a usabilidade e o poder da interação homem-computador para todos os usuários, a partir da exploração do conhecimento dos usuários, tarefas, ferramentas e conteúdo, como também de dispositivos de suporte à interação dentro de diferentes contextos de uso.

Greene (2003) afirma ser possível a construção de um sistema adaptável às necessidades dos usuários em função de informações contextuais em tempo real. Os autores desenvolveram um sistema com *interface inteligente com o usuário* (*Intelligent User Interface - IUI*).

O objetivo final da *interface inteligente* a ser desenvolvida por Greene (2003) é ser adaptável ao usuário, incluindo todas as técnicas que permitem que a interação homem-computador seja adaptada a diferentes usuários e situações de uso. O *Ambient Intelligence System* projetado recolhe informações da forma como os usuários realizaram suas tarefas diariamente e armazenam tais dados em uma rede escrita em Prolog que forma o centro do *Ambient Intelligence Engine* (AIE). O centro tem dois tipos distintos de informação, nodos e conexões. Os vértices representam alguma pessoa, lugar, coisa ou conceito. Conexões ligam um vértice simples a outro vértice. Estas conexões são criadas se dois vértices interagem de alguma forma.

Na atual implementação do AIE tem-se uma aplicação de agenda (*OnTime*) e um aplicativo de e-mail cliente (*SmartMail*). Até agora, estas aplicações têm mostrado bom desempenho, mas ainda há muito trabalho para fazer, segundo os autores, que não explicitam de forma clara em seu artigo as vantagens percebidas nas aplicações de e-mail e de agenda de compromissos.

A idéia de Greene é muito semelhante à apresentada neste trabalho, ou seja, a criação de uma interface que interaja de forma a melhorar a usabilidade com o usuário, sendo que a grande diferença é que a atual proposta é utilizar uma forma para a utilização das atuais ferramentas de desenvolvimento de software para a telefonia móvel, ou seja, agregar um conjunto de ferramentas às atuais que possibilitem uma maneira do aplicativo alterar a interface de acordo com o uso do aplicativo, a maneira proposta também não se baseia em inteligência artificial como em proposto por Greene.

Em Qiu (2004) é apresentado a realização de um estudo empírico objetivando encontrar um método eficiente para projetar interfaces amigáveis para dispositivos com limitações quanto ao tamanho da tela. Neste estudo, foram comparados três modelos para projeto de interfaces:

- (i) *Presentation Optimization* – representa um conjunto de tecnologias cujo objetivo é otimizar o processo de construção de interface de forma que o conteúdo a ser mostrado na tela esteja maximizado e a complexidade de navegação minimizada;
- (ii) *Semantic Conversion* – nesta técnica o texto é convertido de maneira a facilitar a leitura pelo usuário, ou seja, um texto sofre um tratamento e as partes mais importantes são mostradas primeiro, e os textos estão em estruturas que facilitam a navegação e a obtenção de um desejado trecho de um texto;
- (iii) *Zooming* – é baseado nas duas técnicas já apresentadas, ou seja, quando o usuário quer obter informação, o *browser* envia o correspondente dado e mostra na tela de acordo com o tamanho do *display*.

Os experimentos, que contaram com a participação de 27 indivíduos, tiveram o objetivo de avaliar a eficiência e a eficácia destes métodos e utilizaram um simulador de interface para PDA. A análise dos resultados apontou para o método

de *Zooming* como o melhor para projeto de interfaces para pequenos dispositivos. Neste artigo verifica-se a importância e relevância dos projetos de interfaces.

Eisenstein (2001) apresenta um conjunto de técnicas que ajuda ao projetista que está trabalhando no domínio de computação móvel, permitindo ao projetista construir UIs (interface com o usuário) em várias plataformas, respeitando as limitações do dispositivo, ou seja, o aplicativo é desenvolvido para se adaptar ao dispositivo.

O foco do trabalho de Eisenstein se diferencia deste trabalho pelo fato deste focar em dados manipuláveis que construirão a interface com o usuário. Entretanto, o trabalho de Eisenstein assemelha-se ao trabalho proposto nesta dissertação, pois ambos desejam propor técnicas e formas de um aplicativo ser escrito apenas uma vez.

Hilbert (2000) propõe uma maneira de capturar eventos de interface com o usuário automaticamente, tais informações são utilizadas para análise e avaliação de usabilidade. Na presente proposta também foram desenvolvidas técnicas para captura de informações proveniente do dispositivo móvel para uso na construção de interface com o usuário.

Thimbleby (2001) apresenta o conceito de estado de máquina finito (FSM) e analisa um dispositivo móvel representando em com um FSM de 188 estados, montando uma matriz algébrica, que permite análise de usabilidade. Tal artigo assemelha-se a este trabalho por utilizar uma forma matemática de mensurar a usabilidade, na proposta se utiliza de dados para a composição da interface com o usuário.

Palen (2002) discute e analisa a usabilidade de um telefone móvel. Um grupo composto de diferentes classes de pessoas foi estudado inferindo que há uma gama de oportunidade para melhoria da usabilidade. A conclusão apresentada por Palen serve como base para a presente proposta, pois se fundamenta na melhoria da usabilidade de um aplicativo embarcado em um dispositivo móvel.

Após o estudo do atual estado da arte em relação à criação de técnicas para melhoria da usabilidade na telefonia móvel, infere-se que o atual trabalho encaixa-se em um item pouco discutido que é a criação de ferramentas para melhoria da usabilidade, nota-se também que há um grande esforço e concentração nesta área, por ser uma área incipiente e de grande valor comercial e econômico.

## Capítulo 3

# Métodos de Teste de Usabilidade em Aplicações para Telefones Móveis

A usabilidade é, naturalmente, um fator subjetivo. Segundo os critérios estabelecidos pela ISO, há que se analisar a efetividade na execução de uma tarefa específica, como também afirmado por Barbosa (2007). Para tanto existem alguns métodos e métricas para realização desta atividade.

Segundo Rubin (1994), teste de usabilidade é um processo no qual participantes representativos avaliam o grau que um produto se encontra em relação a critérios específicos de usabilidade.

O referido teste pode servir para diferentes propósitos que envolvem tipos de tarefas, medidas de desempenho e disposição de escalas, entrevistas ou inspeções a serem aplicadas, buscando encontrar problemas de usabilidade e fazer recomendações no sentido de eliminar os problemas e melhorar a usabilidade do produto, ou com a finalidade de comparar dois ou mais produtos.

A seguir apresenta-se uma série de conceitos e métodos de teste que foram utilizados neste trabalho para elaboração de uma maneira eficiente para comprovar a eficiência do *método de interação Mandos*.

### 3.1 Método Pensamento em voz alta

Segundo Someren (1994), o método pensamento em voz alta é uma técnica de avaliação de usabilidade que consiste em manter o sujeito do teste falando em voz alta o que quer que lhe venha à cabeça, enquanto executa uma tarefa específica. Diferente de outras técnicas de coleta de dados verbal, não há interrupções ou perguntas sugestivas, de modo que o sujeito é encorajado a contar seus pensamentos sem imaginar interpretação ou explicação do que está fazendo, ele apenas tem que se concentrar na tarefa.

Este método em princípio não leva os distúrbios do processo de pensamento, o sujeito fala enquanto resolve a tarefa quase automaticamente. Os dados coletados são diretos, não há atraso, o sujeito não precisa formar opiniões sobre seus pensamentos, ele os põe para fora conforme vêm a mente. O método do pensamento em voz alta

não precisa sequer ser finalizado, pois o sujeito verbaliza apenas parte de seus pensamentos.

Apesar de ser um método relativamente simples, é bastante aplicável, tanto para observar os critérios de usabilidade, quanto para avaliação do que Someren chama de mapeamento cognitivo, ou seja, o modo de pensar das pessoas. Com este tipo de informação é possível identificar características comuns em grupos de pessoas e assim criar aplicações com melhor usabilidade para um determinado público.

Segundo Ribeiro (1999) há diversas variações do método original, dentre elas a Aprendizagem por co-descoberta, em que dois usuários usam o sistema juntos e capturam as interações um do outro; o Teste Retrospectivo, em que um usuário é convidado a comentar sua atuação gravada em fita de áudio; e Protocolo de Pergunta, em que há uma interação explícita do aplicador do teste e o usuário.

Durante a realização dos testes realizados neste trabalho este método será utilizado em paralelo, incentivando ao usuário que executa os testes pronunciar em voz alta as suas principais dúvidas, dificuldades e sugestões.

### 3.2 Etapas do Método de Medida de Desempenho

Segundo Betiol (2004), o método MUSIC (Método de Medida de Desempenho) (Cooper, 1995; Macleod, 1997) sugere um conjunto de medidas e métricas baseadas nas análises da execução e do resultado da tarefa realizada pelo usuário. As etapas necessárias para avaliar a usabilidade de um produto utilizando o Método de Medida de Performance, ilustrado na Figura 3.1, estão descritas em seguida (Macleod, 1997).

- 1) Definir o produto a ser testado: identificar e descrever o produto, ou os componentes do produto que serão o foco da avaliação de usabilidade.
- 2) Definir o Contexto de Uso: identificar e descrever o contexto em que o produto será usado em função de: características dos atuais ou potenciais usuários, tarefas que serão executadas e ambientes em que os usuários utilizarão o produto.
- 3) Especificar os objetivos e o contexto de avaliação: definir quais requisitos de usabilidade serão analisados e garantir que a avaliação será conduzida em condições tão próximas quanto possível das condições reais de uso do produto. Para isso é necessário que os usuários sejam representativos da população alvo a qual é direcionado o produto, as tarefas sejam representativas em relação ao uso real do produto e as condições da avaliação sejam representativas das condições normais nas quais o produto é utilizado.
- 4) Preparar a avaliação: preparar as tarefas, preparar os usuários, preparar o ambiente do teste.
- 5) Realizar os testes com os usuários: os usuários serão monitorados e registrados por fotografia enquanto executam determinadas tarefas. Ao avaliar a qualidade em uso é importante que os usuários trabalhem sozinhos, tendo acesso somente às formas de ajuda que estariam disponíveis se eles estivessem trabalhando em condições reais de uso.
- 6) Analisar e interpretar os dados: a partir das anotações feitas durante a sessão da análise é possível coletar as medidas e derivar as métricas de usabilidade.

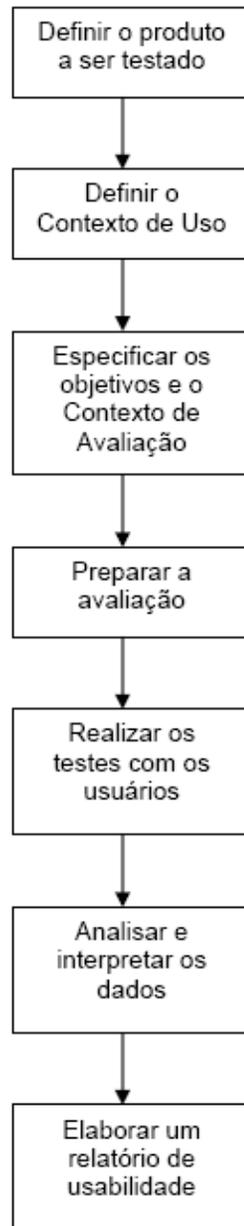


Figura 3.1: Etapas do Método de Medida de Desempenho. Fonte: (Coopers, 1995)

### 3.3 Plano de Teste

O plano de teste consiste na base de todo o teste, especificando como, quando onde, quem, o porquê e o quê sobre o teste de usabilidade. O plano do teste é essencial para a compreensão do teste, sendo um importante veículo de comunicação entre a equipe de desenvolvimento e não deve ser negligenciado (Rubin, 1994).

O formato de um plano de teste depende do tipo de teste e do grau de formalidade do contexto a ser empregado, o Anexo A contém o plano de teste utilizando para validar o método de interação Mandos.

### 3.4 O Avaliador

Segundo Ferreira (2004), o avaliador é responsável por tudo que ocorre durante a sessão de teste. Sua função é interagir com o participante, coletar informações, compilar os resultados dos testes. Ele é responsável por toda preparação para o teste incluindo material, arranjos e coordenação de esforços de outros membros da equipe de teste, se houver. Após a realização do teste, o avaliador interage com os outros membros da equipe de teste, se houver, para assegurar que os objetivos foram alcançados.

O avaliador não deve interferir diretamente no conteúdo das observações coletadas, não deve se comportar como um educador, deve saber quando divergir do plano de teste para alcançar o objetivo de expor as deficiências do produto, deve evitar conflitos com o participante e não deve tirar conclusões precipitadas.

### 3.5 Selecionando os Participantes

A seleção dos participantes é algo crucial para o sucesso efetivo do processo de teste, envolvendo a identificação e descrição de habilidades relevantes e o conhecimento do pessoal que irá usar o produto. A determinação do perfil e caracterização do usuário é determinada nos primeiros estágios do desenvolvimento do produto e servem como base para a seleção dos participantes (Ferreira, 2004).

Uma caracterização genérica do usuário pode envolver, por exemplo:

- Histórico pessoal: idade, gênero, habilidade com computador;
- Histórico educacional: grau, assuntos estudados;
- Experiência computacional: tempo de utilização, frequência, periféricos;
- Experiência com o produto: tempo de utilização, frequência, tarefas;
- Histórico da ocupação: cargo ocupado, responsabilidades, tempo de trabalho na companhia atual.

É interessante determinar um usuário típico, alguém que consiga representar uma média dos usuários finais. Isto auxilia na determinação do teste de usabilidade.

Uma boa maneira de se conseguir uma representação efetiva dos usuários finais é subdividir o perfil do usuário em diferentes categorias, como por exemplo, para um mesmo cargo, subdividir em usuários novatos e em usuários experientes determinando o número de usuários para cada.

### 3.6 Determinando o Número de Participantes

Segundo Betiol (2004), o número de participantes necessários para realizar um ensaio de interação para mensurar a usabilidade é um assunto ainda muito polêmico.

Uma das primeiras pesquisas sobre este assunto foi o trabalho de Virzi (1992) que modelou o número de problemas de usabilidade encontrados em função do número de usuários utilizados na avaliação, chegando a três conclusões:

(a) 80% dos problemas de usabilidade são encontrados por quatro ou cinco participantes;

(b) à medida que aumenta o número de participantes estes revelam cada vez menos informações novas;

(c) os problemas de usabilidade mais graves têm maior probabilidade de serem revelados pelos primeiros poucos participantes.

Nielsen (1993) constatou a partir de vários estudos realizados, que em média 31% dos problemas de usabilidade são encontrados por um único usuário. Com base nesse valor o autor conclui que cinco usuários são suficientes para encontrar 85% dos problemas de usabilidade e que com quinze usuários podem ser encontrados 100% dos problemas.

Betiol (2004) ressalva que, apesar dessas duas pesquisas serem freqüentemente citadas para justificar um pequeno número de usuários nos ensaios de interação, seus resultados são questionados por vários autores (Caulton, 1999; Spool, 2001; Woolrych, 2001).

Em seu estudo Spool (2001) encontrou somente 35% dos problemas de usabilidade com os primeiros cinco usuários. Entretanto, como o próprio autor reconhece, este estudo foi realizado em um ambiente mais complexo em relação aos estudos de Virzi e Nielsen, pois os usuários navegaram em sites Web de *e-commerce* com a tarefa de comprar qualquer coisa que desejassem desde que não ultrapassassem um determinado valor em dinheiro estipulado previamente. As particularidades de cada participante influenciaram claramente estes resultados, pois nenhum usuário comprou o mesmo produto que outro e nenhum percorreu os sites da mesma maneira.

Para Caulton(1999) e Woolrych(2001) o número cinco só pode ser aplicado quando a probabilidade de um usuário encontrar um problema for alta (31% ou mais) e similar para todos os usuários, ou seja, as diferenças individuais entre o perfil dos usuários deve ser mínima, pois quanto mais heterogêneos forem os grupos, mais usuários serão necessários para que todos os subgrupos possam estar representados. Para Caulton (1999) os resultados de Nielsen (1993) podem ser aplicados a testes de usabilidade mais simples que têm como objetivo simplesmente detectar a presença, e não a prevalência estatística, de problemas de usabilidade.

Se os testes forem realizados com poucos participantes muitos problemas sérios da interface podem passar despercebidos, entretanto testar com muitos usuários pode ser inviável, pois cada participante a mais representa custos extras em termos de tempo na execução das sessões e na posterior análise dos dados, e de recursos financeiros para seleção e remuneração do participante e do avaliador (Woolrych, 2001).

### 3.7 Material Utilizado Durante a Sessão de Teste

A seguir, será apresentada uma generalização dos principais materiais necessários à interação com os participantes durante a sessão de teste. O citado material fora baseado e desenvolvido como descrito por Ferreira (2004).

#### 3.7.1 Roteiro do Avaliador

O roteiro do avaliador serve como guia para orientar o avaliador durante a sessão de teste. Bastante semelhante ao plano de teste, descreve o ambiente que será utilizado, as funções do avaliador, o perfil do participante, tarefas do sistema, procedimentos e uma lista dos formulários utilizados. O Anexo B apresenta Roteiro do Avaliador usado neste trabalho.

### 3.7.2. Questionário para Identificação do Perfil do Participante

O questionário para identificação do perfil do participante, completado no início da sessão de teste, relaciona informações históricas sobre o participante auxiliando no entendimento de seu comportamento e desempenho. Está intimamente relacionado à seleção dos participantes através da determinação do perfil do usuário.

Este questionário é em um importante resumo sobre o participante, podendo ser utilizado pelo avaliador ou pelos desenvolvedores que irão observar o teste.

O questionário deve ser facilmente entendido e preenchido, deve ser curto (de uma a duas páginas) e deve se basear em informações extraídas do perfil do usuário que poderão afetar a desempenho dos participantes. O Anexo C apresenta o Questionário para Identificação do Perfil do Participante utilizado neste trabalho.

### 3.7.3. Script de Orientação

Ferramenta de comunicação a ser lida aos participantes antes da realização dos testes. Pode ser lido na própria sala de teste, em uma sala de espera ou em uma sala de reuniões antes de ir para a sala de teste. Descreve o que irá acontecer durante a sessão de teste possibilitando uma visão prévia aos participantes, deixando-os mais à vontade.

Neste documento, deve estar bem claro que o objeto a ser testado é o produto e não o participante.

Este documento deve ter de uma a duas páginas (a não ser que se trate de um teste complexo) e deve utilizar uma linguagem mais profissional, entretanto amigável.

Deve conter uma apresentação sobre o avaliador, sobre o que será realizado, quais os objetivos, a importância e o que se espera do participante, descrição dos equipamentos e recursos utilizados. Deve-se deixar claro que podem ser realizadas perguntas, mas que o avaliador não irá responder a perguntas acerca de tarefas em que o participante é capaz de realizar sozinho de posse dos materiais que já tem disponível e que o avaliador não irá resolver os problemas que forem detectados pelo participante. Deve-se assegurar que os participantes entenderam as orientações passadas. O Anexo D apresenta o Script de Orientação utilizado neste trabalho.

### 3.7.4. Lista de Tarefas

A Lista de Tarefas, também conhecida como Cenário de Tarefas, é a representação do trabalho que os participantes executam utilizando o produto que está com o plano de testes. Descreve os resultados finais que o participante tentará atingir, motivos para executar o trabalho, dados atuais, estados do sistema quando a tarefa for iniciada e mensagens que o participante verá enquanto executa a tarefa. A Lista de Tarefas deve ser distribuída, sendo que a leitura para os participantes é opcional.

A Lista de Tarefas representa a realidade em que será utilizado o produto, portanto se o participante do teste já está familiarizado com o trabalho a ser executado no produto, conseqüentemente se sentirá mais à vontade. O Anexo E apresenta a Lista de Tarefas utilizada neste trabalho.

### 3.7.5. Instrumentos para Coleta de Dados

O propósito dos instrumentos de coleta de dados é registrar acontecimentos ocorridos durante o teste. A intenção é coletar dados concisos e confiáveis durante o teste, podendo ser dados sobre desempenho ou dados preferenciais.

Tanto os dados sobre desempenho, quanto os dados preferenciais podem ser analisados quantitativa ou qualitativamente.

Para se desenvolver este instrumento, os tipos de dados a serem coletados devem ser bem claros e diretamente ligados a questões que necessitam ser resolvidas, extraídas do plano de testes. Deve-se selecionar um método para a coleta de dados que alcance o objetivo do teste.

Há instrumentos para coleta de dados totalmente automatizados capazes de verificar o número e o tipo de batidas de teclas realizadas pelo participante, capazes de detectar quais recursos do programa foram acessados para a realização de uma tarefa e o tempo de acesso em cada recurso, acessos à ajuda on-line etc. Outra maneira é os dados serem coletados de forma manual ou de forma on-line pelo avaliador ou pelo próprio participante e as ações registradas através de um software. Podem ser customizados códigos de até três letras que representam eventos críticos para facilitar no registro das ações. O Anexo F apresenta o formulário de Coleta de Dados pelo Avaliador usado neste trabalho.

### 3.7.6. Questionário de Avaliação do Sistema pelo Participante

Aplicado logo após a realização dos testes, o principal propósito deste questionário é coletar informações preferenciais dos participantes para esclarecer e aprofundar o entendimento do produto apontando pontos fortes e pontos a melhorar, baseando-se nos problemas encontrados.

É importante formular perguntas relacionadas com aspectos difíceis de serem detectados somente com a observação, tais como: sentimentos, opiniões e sugestões para melhoria. Para o desenvolvimento deste questionário, deve-se focar as áreas e tópicos que são necessários atingir, como por exemplo, testes de telas, documentação, painéis de controle, dentre outros. As questões devem direcionar a respostas simples e breves, tornando mais objetivo e facilitando a apuração dos dados, sendo que escalas são muito utilizadas. O teste piloto também pode ser usado para refinar o questionário, pois ele assegura se as questões extraíram as informações corretas. O Anexo G apresenta o Questionário de Avaliação do Sistema pelo Participante utilizado neste trabalho.

### 3.7.7. Tópicos para Questionamento

O objetivo dos tópicos de questionamento é estruturar a sessão de questionamento do participante em relação aos testes realizados. Sugere uma linha de tópicos na qual a natureza exata depende das circunstâncias encontradas durante cada sessão de teste, podendo variar de acordo com determinadas ações tomadas ou comentários realizados pelo participante durante o teste ou de acordo com a experiência que o participante possui.

A realização de anotações durante a sessão de teste sobre itens específicos que necessitam ser aprofundados na sessão de questionamento do participante, é de

grande valia. O Anexo H, Tópicos para Questionamento, apresenta um formulário contendo alguns tópicos que poderão ser discutidos em uma sessão de questionamento do participante. Como estes tópicos são originados principalmente através da observação durante cada sessão de teste, com certeza outros tópicos específicos surgirão com esta observação.

### 3.7.8 Equipamento utilizado para os testes

Para realização dos testes para validar o *Método de interação Mandos* fora utilizado o telefone móvel SL 75, que pode ser visualizado na figura 3.2.



Figura 3.2: Telefone móvel BenQ-Siemens SL75

Os aplicativos utilizados neste teste de usabilidade foram:

- Calculadora – aplicativo que realiza algumas operações aritméticas, ilustrado na figura 3.3, apresenta algumas funções especiais, como por exemplo, armazenar resultados.



Figura 3.3: Calculadora (aplicativo embarcado no telefone SL75)

- Despertador – aplicativo que simula um despertador programável, ilustrado na figura 3.4, além de se programar a hora para despertador, pode-se programar também os dias das semanas, além de outras opções.

As aplicações citadas sem o uso do *método de interação Mandos* estão embarcadas no telefone BenQ-Siemens SL75, originais de fábrica. As aplicações que serão objetos de estudo geralmente são usadas pelo usuário de um telefone móvel esporadicamente e são bastante conhecidas do mesmo, ou seja, o resultado apresentado não se baseará em aplicações que não é utilizada constantemente pelo usuário ou em aplicações em que o usuário praticamente não usa, limitando assim o escopo do resultado a ser apresentado, assim como aplicações cujas tarefas sejam totalmente desconhecidas pelo usuário.

Para se realizar o teste fora feito uma cópia com as mesmas funcionalidades das aplicações usando o *framework Mandos*, que fora desenvolvido para validar o *método de interação Mandos*.



Figura 3.4: Despertador (aplicativo embarcado no telefone SL75)

# Capítulo 4

## Método de Interação Mandos

A proposta deste trabalho consiste na formulação de um método novo de interação com o usuário em aplicativos projetados para serem embarcados em telefones móveis.

O *método de interação Mandos* propõe-se a auxiliar a usabilidade de um aplicativo através de uma nova forma de interação com o usuário, baseado em uma matriz de probabilidades de transição entre tarefas, e apresentar a tarefa em que o usuário tem mais probabilidade em executá-la de acordo como estado atual. Para o melhor entendimento da proposta há a necessidade de explanação de alguns conceitos básicos, que é feito a seguir.

O nome de Mandos para o método de interação é uma referência a um personagem fictício da literatura de Tolkien (2003). Mandos é um *valar* imortal, que jamais se esquece de nada e sabe o que está por vir.

### 4.1 Conceitos Básicos

Apresentam-se em seguida alguns conceitos básicos para o correto entendimento do *método de interação Mandos*.

#### 4.1.1 Tarefa

O *método de interação Mandos* será baseado no conceito de *tarefas*. Uma *tarefa* é uma operação completa iniciada pelo usuário e executada pelo aplicativo. O usuário aguarda uma resposta ou ação, este ato completo, que se inicia quando o usuário inicia o processo de execução que o aplicativo oferece, até o seu final, havendo ou não mudança de tela, este processo definido será chamado de *tarefa*.

Uma *tarefa* sempre deverá ser acionada via um comando emitido pelo usuário.

Este conceito é fundamental, pois a intenção é que o usuário navegue por *tarefas*, sendo que as *tarefas* mais usuais serão mais fáceis de executar, isto é, a quantidade de botões a serem pressionados para executar uma determinada tarefa será inversamente proporcional a sua frequência de uso.

Pode-se citar como exemplo de *tarefas*: sair de aplicação, ativar o alarme de um aplicativo embarcado que simula um despertador, salvar o resultado de uma conta na memória de aplicativo que simula uma calculadora, sair de aplicativo qualquer, etc.

A ordem das tarefas a serem apresentadas aos usuários pode-se dizer que é um processo que evolui no tempo de maneira probabilística, tais processos são denominados Processo Estocásticos. Este processo estocástico pode ser classificado, em relação ao estado, como discreto e, em relação ao tempo, como discreto.

Pode-se dizer que as transições entre tarefas representam um processo markoviano, ou seja, um processo estocástico é dito ser um Processo Markoviano se o estado futuro depende apenas do estado presente e não dos estados passados.

### 4.1.2 Lista de Probabilidades

Com o conceito de tarefa estabelecido pode-se inferir que haverá uma dada probabilidade, após executar uma dada tarefa, de executar uma outra específica, ou seja, existirá uma lista de probabilidades que liga a tarefa que executou com todas as outras tarefas existentes que poderão ser executadas.

Em um aplicativo detentor de um determinado número de tarefas pode-se realizar um estudo e montar uma lista de probabilidades de transição de uma tarefa para todas as outras.

Para melhor esclarecimento, pode-se criar um caso fictício de um editor de texto, após executar a *tarefa copiar* em um texto a probabilidade de executar a *tarefa colar* é alta, este índice será transformado em número através de medições do uso do aplicativo, ou seja, após certo grupo de usuário utilizar por um determinado período este aplicativo, chegará a uma conclusão de que a probabilidade é 0.5, e assim por diante, a intenção é que antes do uso de um dado aplicativo, um ensaio com representantes do público-alvo do mesmo será realizado, com o intuito de coletar um conjunto de aferimentos com objetivo de se determinar um número preciso para a transição de uma específica tarefa com todas as outras tarefas de um aplicativo.

Como já enunciado a ordem de visualização de uma tarefa pode ser considerado um processo Markovino, então a citada lista de probabilidade pode ser denominado com vetor de probabilidade de estado.

Uma lista de probabilidade poderá variar de acordo com uma variável, e uma dentre várias poderá ser utilizada no momento de montar a matriz de probabilidades de transição entre tarefas, que por sua vez será utilizada para montagem da interface com o usuário por um aplicativo.

### 4.1.3 Matriz de probabilidades de transição entre tarefas

Na matriz de probabilidades de transição entre tarefas é onde estará contida toda a informação sobre as transições de estado entre as *tarefas*.

Será uma matriz que representará as *tarefas* pelo índice tanto da coluna como linha, o número de linhas da matriz estará associado ao número de *tarefas* que o aplicativo conterà.

A soma dos valores de uma linha sempre será 1 (um), pois em uma linha conterà a probabilidade de transições entre as *tarefas*.

A probabilidade será relativa à transição das *tarefas* que a linha representa para as *tarefas* que a coluna representa.

Como a citada matriz de probabilidades é formada pelos vetores de probabilidade de estado, esta matriz pode ser estudada como uma matriz de transição de um Processo Markoviano.

Além de podermos utilizar as propriedades de um Processo Markoviano, podemos ainda qualificar o processo de transição entre tarefas como um Processo Markoviano, pois um processo é dito ser uma Cadeia de Markov quando as variáveis randômicas estão definidas em um espaço de estados discreto.

Como um dos estados de transição da matriz de probabilidade é a tarefa sair, a probabilidade de outra tarefa ser executada é zero, ou seja, o processo cessa ao atingir esta condição (estado), temos então um caso especial da Cadeia de Markov, que é conhecida como Cadeia Absorvente de Markov.

A cadeia absorvente de Markov é definida abaixo.

*Definição 1:* um estado absorvente é precisamente aquele no qual há uma probabilidade zero de saída do estado, isto é, uma vez entrado neste estado é impossível deixá-lo e o processo para totalmente ou para e depois recomeça de algum novo estado;

*Definição 2:* a cadeia de Markov é absorvente se:

1. Tem pelo menos um estado absorvente;
2. É possível ir de qualquer estado não absorvente para pelo menos um estado absorvente. Não é necessário fazer essa transição em um passo, nem é preciso atingir cada estado absorvente a partir de um estado não absorvente.

A importância de identificar a matriz de probabilidade como sendo uma cadeia de Markov absorvente é para aproveitar as diversas análises matemáticas já desenvolvidas para futuras análises, como exemplo, se pode conseguir vários tipos de informação importante, partindo da análise deste tipo de cadeia. É possível determinar os seguintes dados:

1. O número esperado de passos antes de o processo ser absorvido;
2. O número esperado de vezes que o processo se encontra em qualquer estado não absorvente;
3. A probabilidade de absorção por qualquer estado absorvente.

A matriz com as probabilidades será armazenada em um arquivo XML. Cada tarefa conterá as probabilidades de transição para as outras, quando não houver a possibilidade de transição entre duas tarefas a probabilidade será 0 (zero), sendo assim a interface que representará tal tarefa não será exibida ao usuário.

Haverá dois tipos de matriz, a matriz estática e a matriz dinâmica que será detalhado a seguir.

#### **4.1.2.1 Matriz Estática**

A matriz estática possibilitará ao aplicativo apenas ler as informações, não podendo inserir informações capturadas do aplicativo, ou seja, uma vez preenchida a matriz o aplicativo que a usará sempre irá obedecer àquelas probabilidades.

#### **4.1.2.2 Matriz Dinâmica**

Este tipo de matriz de probabilidades de transição entre tarefas permitirá que os dados oriundos do aplicativo sejam inseridos na matriz, ou seja, a probabilidade de transição entre tarefas irão se modificando de acordo com o uso do aplicativo pelo usuário.

Neste caso, como será feito adquirido dado do usuário, não poderemos mais afirmar que haverá uma probabilidade exata de mudança de estado e sim uma medição da frequência relativa, que fornece um indicar preciso proporcional à quantidade de dados colhidos, ou seja, quanto mais dados de uso do aplicativo por um determinado usuário, maior a precisão da frequência relativa.

#### 4.1.2.2.1 Frequência relativa

Para deixar mais claro a idéia de frequência relativa, utiliza-se a seguinte suposição: sendo  $A$  e  $B$  dois eventos associados a um experimento, ensaiamos  $n$  vezes tal experimento. Denotemos por  $n_A$  o número de vezes que o evento  $A$  ocorre, e chamemos de  $n_B$  o número de vezes que o evento  $B$  ocorre nas  $n$  repetições. A frequência relativa do evento  $A$  é dada por  $f_A = n_A / n$ .

A frequência relativa apresenta propriedades semelhantes às que veremos adiante para probabilidades:

- $0 \leq f_A \leq 1$
- $f_A = 1$  se  $A$  ocorre em todas as  $n$  repetições;
- $f_A = 0$ , se  $A$  não ocorre em nenhuma das  $n$  repetições;

Se considerarmos  $f_A$  uma função de  $n$ , verificamos que "converge" para  $P(A)$  quando fazemos  $n \rightarrow \infty$ .

A última propriedade não é difícil se compreender intuitivamente, que a frequência relativa de um evento vai se estabilizando e varia cada vez menos à medida que aumentamos o número de ensaios. Esta característica é chamada de regularidade estatística.

A frequência relativa pode fornecer um número bastante preciso para sabermos a chance de ocorrência de um evento  $A$ . À medida que se aumenta o número de ensaios daquele experimento, sabe-se que ele se estabilizará próximo de um número, a probabilidade do evento  $A$  ocorrer. Contudo, não está claro o quão grande deve ser  $n$  para que se possa obter a precisão desejada.

#### 4.1.4 Estudo do tempo de aprendizagem do usuário

A matriz dinâmica oferecerá a funcionalidade de adaptabilidade do aplicativo em concordância com as tarefas mais usadas pelo usuário, possibilitando que *tarefas* mais usuais sejam mais fáceis de serem ativadas e com isto oferecendo ao usuário um aplicativo mais ágil e adequado ao seu estilo, mas para tanto é necessário que haja um estudo de como e quando adquirir os dados do aplicativo.

No princípio do uso de um aplicativo haverá um processo de aprendizado, ou seja, não necessariamente as tarefas mais utilizadas neste período serão as mais utilizadas após o encerramento do mesmo.

Não se propõe neste trabalho a definição de uma maneira de se obter o valor exato do tempo de aprendizado de um dado aplicativo, mas o *framework Mandos*, que será implementado para validar o *método de interação Mandos*, já irá, de antemão, prever esta situação e está preparada para isto, destaca-se que este tempo possa ser medido pela quantidade de botões pressionados, pelo tempo em segundos de uso do aplicativo ou quantidade de *tarefas* acessadas. Este valor estará contido no arquivo texto onde estará contida a matriz de probabilidades de transição entre tarefas de um dado aplicativo.

### 4.1.5 Exemplo de uma matriz de probabilidades de transição entre tarefas

Para preencher uma matriz de probabilidades de transição entre tarefas é necessário que haja um determinado número de *tarefas*, que serão definidas a seguir.

O exemplo de montagem da matriz de probabilidades de transição entre tarefas será de um simples despertador.

- Tarefa 01 – Ativar um despertador
- Tarefa 02 – Desativar um despertador
- Tarefa 03 – Ajustar a hora
- Tarefa 04 – Sair do aplicativo

Com isto podemos montar uma matriz quadrada que conterà a probabilidade de mudança entre as *tarefas*. As probabilidades utilizadas nesta matriz são fictícias, a obtenção destes valores será determinada pelo uso de um específico grupo de usuários, que representarão um determinado público alvo, não esta no escopo deste trabalho definir ou estabelecer o melhor meio de obter tais valores, apenas de utilizá-lo e estar ciente de que é inteiramente possível e viável obtê-los.

	<b>Linha</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>
<b>Coluna</b>	<b>Tarefas</b>	<i>Ativar o despertador</i>	Desativar o despertador	<i>Ajustar a hora</i>	<i>Sair do aplicativo</i>
<b>01</b>	<i>Ativar o despertador</i>	0	0.1	0.4	0.5
<b>02</b>	<i>Desativar o despertador</i>	0.1	0	0.1	0.8
<b>03</b>	<i>Ajustar a hora</i>	0.5	0.1	0.2	0.2
<b>04</b>	<i>Sair do Aplicativo</i>	0	0	0	1

Tabela 4.1: Matriz de probabilidades de transição entre tarefas

Observe que o primeiro ato está localizado na coluna, o ato posterior está na linha. Exemplificando: A probabilidade de *Ativar um despertador* (tarefa 01, verifique que está na linha 1) e logo após *Ajustar a Hora* (tarefa 03, que está na 4 coluna) é de 0.5. Note para melhor compreensão as probabilidades relativas às *tarefas* estão sombreadas. A matriz de probabilidades de transição entre tarefas é composta das citadas probabilidades de transição entre as *tarefas* e as probabilidades relativas ao estado inicial e final.

Quando a probabilidade for 0 (zero) a interface que representa a *tarefa* não estará disponível para o usuário, ou seja, realmente a probabilidade de acionar tal tarefa será nula.

Optou-se por armazenar esta matriz em um arquivo XML, por ser um formato simples de entendimento para o ser humano, auxiliando no objetivo de se obter uma forma simples de um não especialista em programação poder alterar a interface com o usuário de um aplicativo. Mas detalhes sobre o formato XML está contido no Capítulo 6, Framework Mandos.

## 4.2 Níveis de uso da Matriz de probabilidades de transição entre tarefas

Há três níveis de uso da matriz de probabilidades de transição entre tarefas, que está listado a seguir:

### *1.º Nível – inicialização estático*

Neste tipo o aplicativo apenas consultará a matriz de probabilidades de transição entre tarefas na inicialização do aplicativo e as *tarefas* no aplicativo não se adequarão, de acordo com o presente estado, permanecerão estáticas, ou seja, manterão suas posições originais.

Este nível de uso é apenas para diferenciar as posições das *tarefas* de um aplicativo em sua posição inicial, mantendo o estilo de navegação e usabilidade tradicional.

Este nível resolve o problema de versionamento de código, ou seja, a interface de um determinado aplicativo poderá ser alterada apenas modificando os dados de um arquivo externo, facilitando a criação de várias versões de acordo com o público alvo de um telefone móvel.

### *2.º Nível – inicialização com transição dinâmica das tarefas*

Neste nível de uso a matriz será consultada apenas na inicialização do aplicativo, mas todos os seus dados serão utilizados.

O aplicativo irá se adequar dinamicamente de acordo com o corrente estado, alterando todos os acessos das tarefas, privilegiando as tarefas que tenham a maior probabilidade de serem invocadas.

### *3.º Nível – adaptação da matriz de transição*

Com a captura de informações proveniente do uso de um aplicativo pelo usuário e inserção destes dados convertidos em probabilidade na matriz de probabilidades de transição entre tarefas, aperfeiçoa-se o processo de adequação do aplicativo a um determinado estilo, pois com a inserção de dados do usuário cria-se um perfil próprio do usuário, privilegiando o acesso de determinadas tarefas mais usuais a partir do uso pessoal.

## 4.3 Ilustrando o uso do Método de Interação Mandos

A Figura 4.1 ilustra um aplicativo que tem como função de ser utilizado como uma calculadora.



Figura 4.1: Calculadora

O aplicativo citado pode ser encontrado no celular BenQ-Siemens SL75, o mesmo já vem embarcado de fábrica, para acessá-lo seguir a seqüência: Menu, Extras, Calculadora.

A calculadora apresenta algumas opções a serem utilizadas pelo usuário, como se pode visualizar na Figura 4.2.

As opções apresentadas no aplicativo citado sempre se manterão estática, ou seja, a ordem será sempre a mesma.

O método de Interação Mandos propõe justamente uma nova interação com o usuário, ou seja, após o usuário utilizar, por exemplo, a função salvar na memória, a próxima função a ser utilizada, dentre as apresentadas, hipoteticamente seria abrir a memória, então a mesma função deveria ser uma das primeiras a ser apresentada na lista de opções, como ilustrado na Figura 4.3.



Figura 4.2: Opções da calculadora

Note que na Figura 4.3, a ordem de apresentação das funções da calculadora são dinâmicas, ou seja, mudam de acordo com o corrente estado da calculadora, o estado é alterado toda vez que uma função é utilizada, tentando através de observações apresentar sempre uma lista de opções de maneira que o usuário tenha que despende um menor esforço para alcançar o que deseja.



Figura 4.3: Calculadora utilizando o Método de Interação Mandos.

#### 4.4 Hipóteses a serem avaliadas sobre o Método de Interação Mandos

Como o objetivo do método de interação proposto é a melhoria da usabilidade nos aplicativos que em seu desenvolvimento o utiliza, a análise do mesmo é realizada com base nos conceitos de usabilidade, verificando se houver aprimoramento na usabilidade. Para tanto, se escolheu um dos conceitos apresentados para realizar a análise, o conceito de Nielsen foi escolhido por apresentar um maior número de aspectos que define a usabilidade, o que auxilia na análise da abrangência e eficácia do método de interação.

- **Intuitividade:** O *método de interação Mandos* propõe uma nova maneira de interação com o usuário, o que pode causar uma rejeição inicial do usuário experiente ao aplicativo. A intenção é que com o uso constante do método de interação e a passagem do primeiro impacto o usuário tende a ver o aplicativo que utilize o *método Mandos* mais intuitivo devido ao fato da interface com o usuário ser montada de acordo com o estado corrente, o que tende a facilitar o entendimento do aplicativo.

Como o experimento feito neste trabalho não poderá avaliar o uso do aplicativo que utiliza o método Mandos em longo prazo, estabelece-se a:

**Hipótese I** – A avaliação dos aplicativos que não utilizam o *método de interação Mandos*, no que se diz respeito à intuitividade, será melhor do que a dos aplicativos que utilizam o citado método de interação.

- **Eficiência:** Pode-se inferir que a eficiência é aumentada consideravelmente, pois o número de botões para ter acesso a uma determinada tarefa é inversamente proporcional à probabilidade de uma tarefa ser selecionada pelo usuário, ou seja, o sistema visa ao alcance de uma alta produtividade do usuário. Com base nisto se estabelece a:

**Hipótese II** – A avaliação dos aplicativos que utilizam o método de interação Mandos, no que se diz respeito à eficiência, será melhor do que a dos aplicativos que não utilizam o citado método de interação.

- **Memorização:** Como a construção da interface do usuário será baseado no estado corrente da aplicação e as tarefas mais utilizadas serão apresentadas mais facilmente acredita-se que isto facilite a memorização do usuário, pois as seqüências de tarefas mais comuns sempre serão mais fácil de se executar. Com base nisto se estabelece a:

**Hipótese III** – A avaliação dos aplicativos que utilizam o método de interação Mandos, no que se diz respeito à memorização, será melhor do que a dos aplicativos que não utilizam o citado método de interação.

- **Redução de Erros:** Sendo as tarefas mais usuais mais fácil de serem alcançadas, acredita-se que isto irá resultar em um aprimoramento na redução de erros. Com base nisto se estabelece a:

**Hipótese IV** – A avaliação dos aplicativos que utilizam o método de interação Mandos, no que se diz respeito à redução de erros, será melhor do que a dos aplicativos que não utilizam o citado método de interação.

- **Satisfação:** O método de interação Mandos traz uma nova forma de interação com o usuário facilitando o uso do aplicativo o que espera que se resulte em uma satisfação do usuário. Com base nisto se estabelece a:

**Hipótese V** – A avaliação dos aplicativos que utilizam o método de interação Mandos, no que se diz respeito à satisfação, será melhor do que a dos aplicativos que não utilizam o citado método de interação.

Neste trabalho estabelece-se que se em 4 ou mais dos 5 quesitos dos aplicativos que utilizarem o *método de interação Mandos* apresentarem uma avaliação superior, a hipótese principal é dada como verdadeira, pois com este índice tem-se a maioria absoluta.

**Hipótese Principal** – O uso do *método de interação Mandos* em aplicativos embarcados em telefones móveis resultará em um aprimoramento da usabilidade.

# Capítulo 5

## Framework Mandos

Para validação do *método de interação Mandos*, proposto neste trabalho, desenvolveu-se o *framework Mandos* que emprega a idéia anteriormente de interação, para assim poder implementar alguns aplicativos que usem o *método de interação Mandos* e assim poder mensurar o quanto e se aprimorou a usabilidade do aplicativo.

### 5.1 Frameworks de Desenvolvimento de Software

Neste trabalho define-se *framework* como sendo um conjunto de classes-base que serão utilizadas para a criação de aplicativos. Nesta monografia, o *framework* proposto disponibilizará classes para construir UI utilizando a arquitetura J2ME de maneira que os aplicativos desenvolvidos tenham algumas características que aperfeiçoem a sua usabilidade.

Outra definição para *framework*, citada a seguir, por Larman (2000):

Framework é como um subsistema extensível para um conjunto de serviços relacionados. Também é possível ser classificado como um conjunto coeso de classes que colaboram para fornecer serviços para o núcleo invariante de um sistema lógico. Contém classes concretas e abstratas que definem interfaces a serem seguidas, interações entre objetos das qual as aplicações devem participar como também outros invariantes.

Obedecendo a Padrões de Projeto, pode-se construir *frameworks* que auxiliem o desenvolvedor no reuso de análise, projeto e codificação. Pois, segundo Reinaldo (2003), utilizando métodos que já foram desenvolvidos e depurados, o tempo de desenvolvimento, de testes e as possibilidades de erros na nova produção se tornam menores.

Em relação ao reuso de código, em Silva (2000) está exposto que um programador que usa linguagem C, por exemplo, utiliza-se de bibliotecas de funções. Isto se classifica como reutilização de rotinas. A reutilização no nível de módulo corresponde a um nível de granularidade superior à reutilização de rotinas. A reutilização de classes em orientação a objetos, segundo Meyer (1988), corresponde à reutilização de módulo. Quando uma classe é reutilizada, um conjunto de rotinas é reutilizado (métodos), bem como uma estrutura de dados (atributos). Reutilizar

classes, portanto, tende a ser mais eficiente que reutilizar rotinas e de mais alto nível de granularidade (uma classe de objetos tende a ser um artefato de software mais complexo que uma rotina isolada).

Segundo Reinaldo (2003), a fase do projeto para construção de *frameworks* deve produzir uma arquitetura bem estruturada através da padronização de classes com fácil integração, portabilidade e consistência. Para se ter um projeto de *frameworks*, necessita-se de três itens:

- o desenvolvedor, que é peça fundamental, será encarregado de manter, organizar e escolher as classes que irão compor a estrutura do *framework*;
- a definição do domínio, que leva ao desenvolvedor decidir que tipo de ferramenta é possível gerar com a estrutura do *framework*;
- futuras aplicações, que possibilita que diferentes ferramentas possam ser criadas enfocando uma mesma base de estrutura do *framework*.

Segundo Silva (2000), pode-se afirmar que o desenvolvimento de um *framework* é mais complexo que o desenvolvimento de aplicações específicas do mesmo domínio, devido a necessidade de considerar os requisitos de um conjunto significativo de aplicações, de modo a dotar a estrutura de classes do *framework* de generalidade, em relação ao domínio tratado; além da necessidade de ciclos de evolução voltados a dotar a estrutura de classes do *framework* de alterabilidade e extensibilidade.

Em relação a sua utilização, Fayad (1997) classifica em dois tipos: caixa branca e caixa preta.

O *framework* Mandos proposto é do tipo caixa preta, que segundo Reinaldo (2003) o aplicativo que irá usá-lo utiliza as funcionalidades existentes no *framework*, sem a possibilidade da arquitetura ser vista ou alterada. A extensão é feita a partir de interfaces fornecidas e definidas para os componentes. As composições e instanciações definem as particularidades desta aplicação. Através de padrões definidos no projeto, os recursos existentes são reutilizados e estendidos por meio da definição e integração de componentes que se ajustam a uma interface específica.

Há três modelos onde o *framework* pode ser utilizado: h: *framework* de suporte, *framework* de aplicação e *framework* de integração. Como descrito por Fayad (1997), *framework Mandos* enquadra-se no tipo de *framework* de aplicação, que segundo Reinaldo (2003), trata de assuntos que envolvam o projeto de construção de aplicações específicas. Encapsula conhecimento aplicável a uma vasta gama de aplicações, por exemplo, construção de interface gráfica para usuário, telecomunicações, finança, produção e educação, geoprocessamento, etc.

## 5.2 Trabalhos correlatos em desenvolvimento de frameworks

Kim (2002) propõe um *framework* para estudar a relevância do contexto de uso na Internet Móvel. Segundo os autores, ainda não foram identificados os locais e situações onde a Internet Móvel tem sido utilizada frequentemente e quais os impactos dos contextos no uso destes dispositivos.

Os autores definem contexto móvel como qualquer informação pessoal e ambiental que pode influenciar o comportamento da pessoa durante a utilização da

Internet Móvel, ou seja, tantos estados emocionais e físicos quanto à localização do usuário e a quantidade de pessoas no ambiente podem causar impacto no uso.

Segundo Shiple (2006), para investigar os diferentes tipos de problemas de usabilidade que podem ser causados pelos diferentes contextos, os autores desenvolveram, inicialmente, um *framework* para classificar os problemas de usabilidade dentro de categorias. Então eles criaram o *Mobile Information Architecture*, uma extensão do *Information Architecture*, um conhecido *framework* para melhorar a qualidade das experiências dos clientes na Internet não móvel. Esta nova arquitetura é composta de 4 elementos: Representação, Estrutura, Navegação e Conteúdo.

Participaram do estudo 37 indivíduos, cada um dos quais recebeu um telefone celular para utilizá-lo durante 2 semanas. Ao utilizarem a Internet móvel, era necessário também preencher um formulário sobre a tarefa realizada, uma espécie de diário de bolso. A análise dos diários foi realizada sob dois aspectos:

- (i) Contexto de uso;
- (ii) Problemas de usabilidade.

Os resultados do estudo indicam 3 importantes constatações em termos do contexto móvel e problemas de usabilidade:

- (i) As pessoas não utilizam a Internet móvel em todo contexto possível;
- (ii) A disponibilidade das mãos, o movimento das pernas e o nível de distração têm um impacto significativo no uso da Internet Móvel;
- (iii) Diferentes problemas de usabilidade são causados mais freqüentemente por diferentes contextos de uso.

Os resultados obtidos no estudo de Shiple têm várias implicações positivas de perspectiva prática e teórica. Teoricamente, o estudo fornece um *framework* de contextos de uso e problemas de usabilidade na Internet Móvel. De ordem prática, os resultados dos estudos indicam que a Internet Móvel é mais usada em poucos contextos.

### 5.3 Arquitetura J2ME (Java 2 Micro Edition)

O *framework Mandos* será projetado para ser usado em conjunto com a arquitetura J2ME, apresentando uma nova maneira de estruturar a interface com o usuário. Para compreender a arquitetura J2ME, segue-se uma breve apresentação da mesma.

Segundo Feijó (2001), atualmente existem três edições diferentes da linguagem Java – *Enterprise Edition* (J2EE), *Standard Edition* (J2SE) e *Micro Edition* (J2ME), cada uma sendo direcionada para diferentes categorias de dispositivos. A linguagem Java 2 *Micro Edition* (J2ME) é uma simplificação da linguagem Java 2 *Standard Edition* (J2SE) realizada através da remoção e modificação de partes fundamentais de J2SE com o objetivo de criar um ambiente de execução de programas para dispositivos com grandes restrições de memória e processamento.

#### 5.3.1 Definição da Arquitetura J2ME

Uma grande parte do trabalho de definição da arquitetura da nova linguagem foi dedicada a considerar quais componentes da linguagem Java original deveriam ser cortados, mantidos ou modificados. Com a observação da variedade de dispositivos envolvidos, o J2ME adotou uma arquitetura modular e escalável, consistindo de um

conjunto de blocos que se complementam para representar as particularidades dos grupos de dispositivos existentes.

O J2ME tem como alvo dois grupos distintos de produtos:

- Dispositivos pessoais, móveis e conectados à informação (information connected): telefones celulares, pagers e PDAs são os melhores exemplos para esta classe
- Dispositivos compartilhados, fixos e conectados à informação (information connected): exemplos típicos são as Internet TVs, telefones com Internet, comunicadores high-end e sistemas de navegação de carros.

Embora todos esses dispositivos tenham muitas coisas em comum, eles também diferem na forma, função e características. Para isso existe o conceito de configuração. A especificação de uma configuração mínima em termos de hardware e bibliotecas padrão para o dispositivo, que é a Connected, Limited Device Configuration (CLDC), é prevista para a primeira categoria de produtos, e a Connected Device Configuration (CDC), para a segunda categoria de dispositivos (UFRGS, 2006).

Há ainda o conceito de perfil (profile). Um perfil é um conjunto de bibliotecas que são muito mais específicas a uma categoria de dispositivos do que as bibliotecas disponíveis pela configuração. Perfis são implementados ao topo da configuração.

O J2ME define três camadas de software construídas sobre o sistema operacional presente em cada dispositivo, a saber (Pessoa, 2003):

- Camada da Máquina Virtual Java (JVM), que é uma implementação da máquina virtual Java baseada no sistema operacional presente em cada dispositivo;
- Camada de definição de uma Configuração, que define uma plataforma mínima para uma larga categoria de dispositivos;
- Camada de definição de um Perfil (profile), mais próxima dos usuários e desenvolvedores de aplicações, que define a API voltada para demandas específicas de um segmento de mercado, tais como, carros, celulares, televisores, etc.

Como ilustrado na Figura 5.1, no nível mais abaixo está o hardware, e logo acima dele o seu sistema nativo. Após, vem o CLDC (Connected Limited Device Configuration) que é onde está a Máquina Virtual Java (KVM). O CLDC tem como objetivo definir uma plataforma Java padrão mínima para pequenos dispositivos. A KVM irá permitir que APIs Java de alto nível sejam construídas (UFRGS, 2006).

As APIs MIDP (Mobile Information Device Profile) rodam em cima do CLDC.

O MIDP é a definição de uma arquitetura e APIs associadas necessárias para prover um ambiente de desenvolvimento aberto para MIDs (mobile information devices).

As APIs MIDP (Mobile Information Device Profile) rodam em cima do CLDC.

O MIDP é a definição de uma arquitetura e APIs associadas necessárias para prover um ambiente de desenvolvimento aberto para MIDs (mobile information devices).

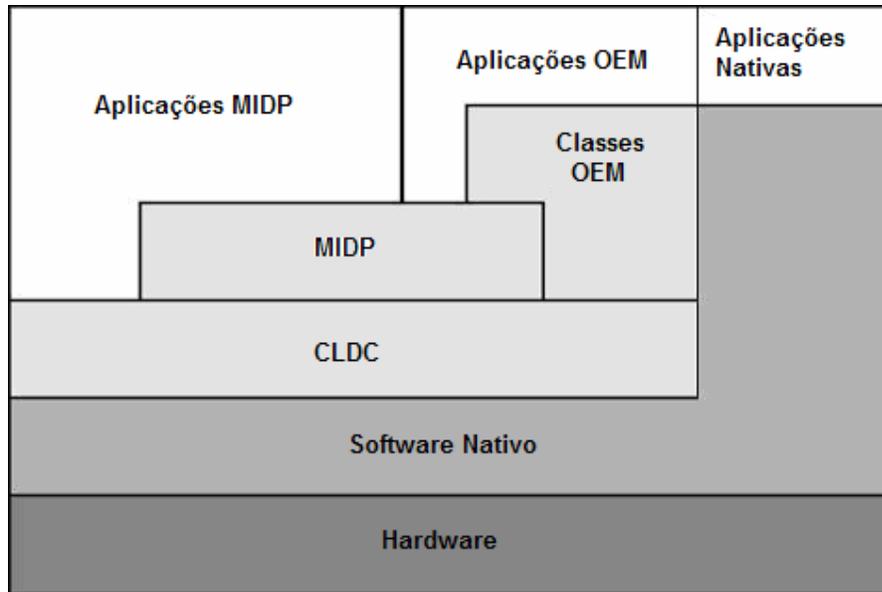


Figura 5.1: Camada MIDP e CLDC, baseado em UFRGS (2006)

As APIs MIDP (Mobile Information Device Profile) rodam em cima do CLDC.

O MIDP é a definição de uma arquitetura e APIs associadas necessárias para prover um ambiente de desenvolvimento aberto para MIDs (mobile information devices).

Quanto aos aplicativos, um MIDlet é aquele que usa APIs definidas pelo MIDP e CLDC, e são portáveis entre vários aparelhos. MIDlet é um aplicativo em Java para dispositivos móveis, mais especificamente para a máquina virtual J2ME. Em geral são aplicações para serem executadas em celulares, como jogos entre outros.

Uma aplicação OEM são aquelas que não fazem parte da especificação. Já as nativas são as que estão implementadas diretamente no sistema e não são escritas em Java.

Baseado nesse conceito de camadas, apenas as funcionalidades comuns a todos os dispositivos fariam parte das camadas mais centrais, enquanto que as particularidades ficariam restritas apenas aos perfis e configurações mais específicas. Naturalmente, um programa escrito para um dado perfil tem a portabilidade garantida para qualquer dispositivo que suporte tal perfil. A idéia é que isso ocorra da mesma maneira que aplicações para PCs são desenvolvidas para um determinado sistema operacional sem preocupação com o fabricante do computador que irá executá-lo. Atualmente existem duas configurações, uma para dispositivos fixos com grande demanda de energia e conexão, *Connected Device Configuration* - CDC, e outra para dispositivos móveis e mais simples, *Connected Limited Device Configuration* - CLDC. Um perfil que abrange celulares e *paggers*, *Mobile Information Device Profile* - MIDP já está definido sobre o CLDC. Da mesma forma, já existe um perfil sobre o CDC, o *Foundation Profile*. Outros perfis estão em desenvolvimento, como é o caso do *Personal Profile* (extensão do *Foundation Profile*).

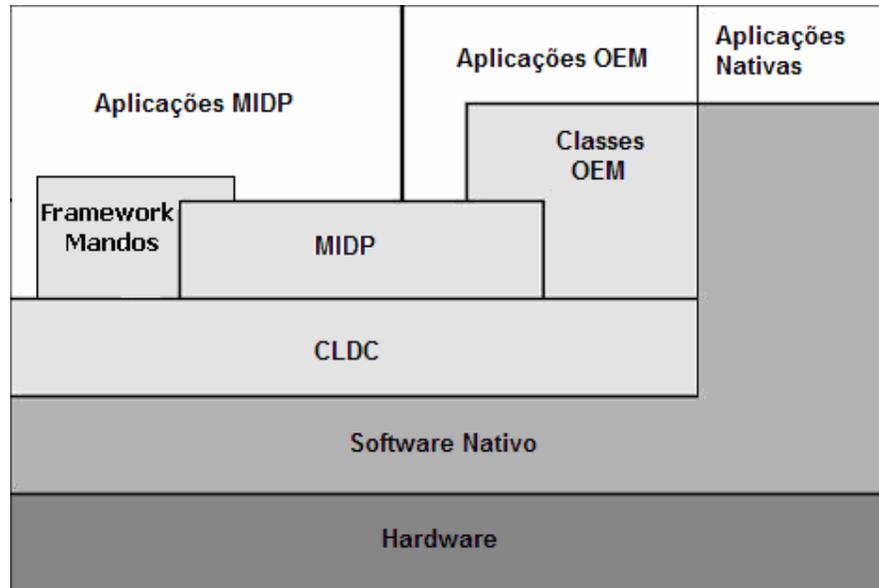


Figura 5.2: Camada do *Framework Mandos* entre as Camadas MIDP e CLDC, baseado em UFRGS (2006)

Na Figura 5.2, ilustra-se onde está o *framework Mandos* entre as camadas apresentadas. Note que o mesmo utiliza a camada CLDC e MIDP, e somente as aplicações MIDP podem utilizá-lo.

### 5.3.2 Breve visão geral da Arquitetura J2ME

Segundo Martins (2002), toda comunicação tem que acontecer em ambas as direções, ou seja, ir e vir, para ser eficiente. Aplicações MIDP não é exceção nenhuma. O MIDlet provê um conjunto de métodos que você pode usar para comunicar-se com o gerenciador de aplicações (Muchow, 2002):

- **NotifyDestroyed** informa ao gerenciador que a sua aplicação quer ser terminada. Ao chamar este método, o resultado não será ter o método "destroyApp" ser executado, portanto você terá de chamá-lo manualmente.
- **NotifyPaused** notifica o gerenciador que a sua aplicação quer ser pausada. ResumeRequest solicita que o gerenciador volte a rodar a sua aplicação, se este estiver pausada.
- **GetAppProperty** busca as informações de configuração de uma aplicação. Tal informação pode ser acessada através do fornecimento de um manifesto ou um arquivo de descrição de um aplicativo (application descriptor file). Em outras palavras, este método dá acesso a um arquivo de inicialização privada.

Segundo Fonseca (2005), na classe MIDlet podemos encontrar um método para cada mudança de estado de seu aplicativo. Esses métodos devem ser implementados sempre na classe principal de seu aplicativo, pois esta classe deverá estender a classe MIDlet.

Assim que o usuário fizer uma chamada a um MIDlet, este aplicativo usará o método `startApp()` para mudar o estado deste MIDlet, neste caso para ativo. Vamos supor agora que com o aplicativo no estado ativo, o usuário receba uma ligação.

Neste momento o método `pauseApp()` será invocado para que o estado do aplicativo seja alterado para interrompido, e assim que esta ação for concretizada o método `notifyPaused()` será chamado. Após finalizar a ligação o usuário poderá voltar ao aplicativo que estava interrompido. Neste momento o método `startApp()` modifica o estado do aplicativo que estava interrompido, voltando agora para ativo.

Após a mudança do estado, o método `resumeRequest()` será chamado. E finalmente quando o usuário deseja fechar o aplicativo, o método `destroyApp()` passa o MIDlet para o estado destruído. Quando ocorre esta mudança o método `notifyDestroyed()` é invocado. No caso em que ocorre algum problema na mudança entre estados, a Exception `MIDletStateChangeException` é lançada. Quando isso ocorre o estado da aplicação é imediatamente alterado para destruído.

Muitos desenvolvedores se equivocam ao achar que o estado inicial de um MIDLET é o estado ativo, o que é normal, pois é o mais lógico. Porém o estado inicial do MIDLET é o interrompido (pausado) como mostro na Figura 5.3.

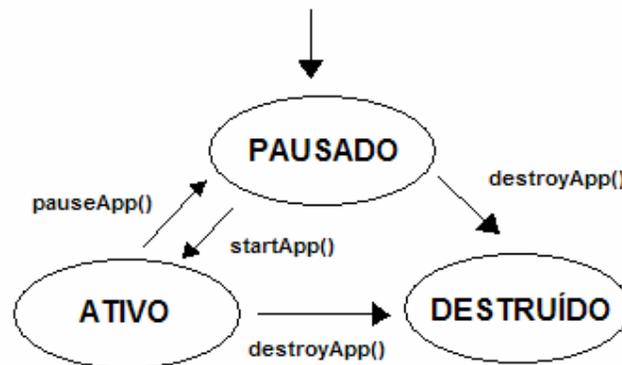


Figura 5.3: Ciclo de vida de um MIDlet

Sobre os componentes gráficos para construção da interface com o usuário, pode-se dizer que há basicamente três famílias, que estão definidas no pacote "lcdui" e são agrupados baseados em suas classes base: Componentes de tela; Componentes de item; e Componentes diversos de display.

Os componentes de tela são descendentes da classe abstrata "*Screen*" e provêm os *widjets* de interface gráfica em forma de janela. O objeto "*Form*" (formulário) é um descendente da classe "*Screen*", que contém e forma controles de interface gráfica. Outros componentes de "*Screen*" incluem alertas (*Alert*), caixas de diálogo (*dialog box*), listas (*List*) e caixas de texto (*TextBox*), sendo esta última uma forma de entrada de texto que suporta múltiplas linhas.

Os componentes de item são controles tradicionais, estes controles todos são descendentes da classe "*Item*". Esta classe provê uma API uniforme para se colocarem etiquetas de nome, tratamento de eventos, e exibição de controles. *ChoiceGroup*, *DateField*, *Gauge*, *ImageItem* e *StringItem* são outros componentes da classe *Item*.

Todos os componentes diversos de display são, na sua grande maioria, descendentes da classe de hierarquia mais alta e abstrata "*Displayable*". Este grupo inclui componentes como a classe "*Command*", que integra os botões de comando; "*Ticker*", que integra as caixas de texto rolantes; "*Graphics*", que exhibe os gráficos; e

"Choice", que é uma interface de manipulação de seleções pré-definidas, que, por sua vez, não se encaixam em qualquer outra categoria.

Toda a parafernália gráfica é gerenciada por um objeto "Display", a partir do qual, cada aplicação tem acesso a uma única e privada instância. Esta instância pode ser obtida através do método estático "Display.getDisplay". Além dos métodos para concentrar o foco da tela em um elemento em particular (setCurrent) e descobrir qual o elemento com o foco (getCurrent), a classe "Display" também expõe vários outros métodos muito úteis para obtenção de informação sobre as capacidades do display do dispositivo. Entre estas capacidades estão a detecção de suporte à cores (isColor) e quantas cores são suportadas (numColors).

A hierarquia entre os componentes gráficos da arquitetura J2ME é ilustrado na Figura 5.4.

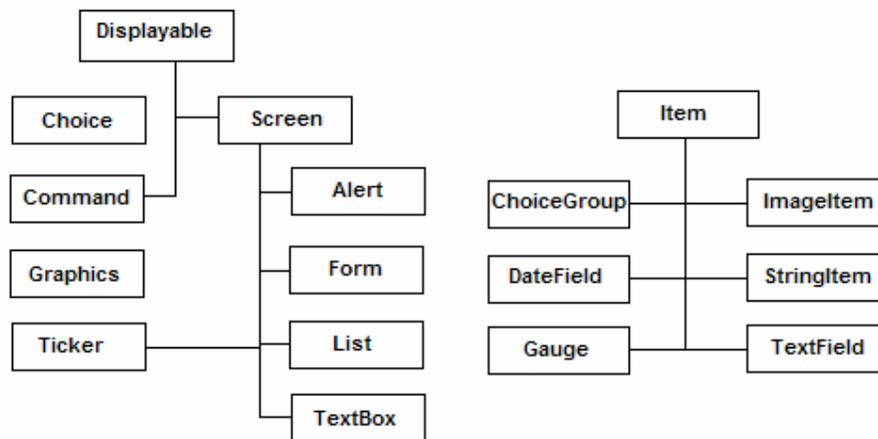


Figura 5.4: Hierarquia entre os componentes da arquitetura J2ME, baseada de (Martins 2002)

### 5.3.3 Classe *Command* da Arquitetura J2ME

O estudo de caso que será descrito tem por objetivo apresentar uma nova forma de utilizar a classe *Command*, exemplificando o uso do *framework Mandos*. Para tanto é necessário uma breve descrição da classe *Command*.

Segundo Muchow (2002), um objeto *Command* contém informações sobre um evento. O modo mais simples de pensar em um objeto *Command* é como um “botão”, algo que você pressiona ou seleciona. A Figura 5.5 mostra dois objetos *Command*, um com o rótulo “Sair” e outro com o rótulo “Opções”, que por sua vez contém uma lista de diversos outros *Commands*. Esse objeto *Command* está associado ao botão imediatamente abaixo dele no dispositivo.



Figura 5.5: Calculadora no modo expandido (aplicativo embarcado no celular BenQ-Siemens SL75).

A Figura 5.6 ilustra o *menu de opções* quando selecionado, mostrando todos os *Commands* que foram adicionados e representam diversas funções do aplicativo calculadora.

A idéia de que um objeto *Command* é um botão representa um bom paralelo.

Na arquitetura J2ME para processar eventos exige um trabalho antecipado:

1. Criar um objeto *Command* para conter as informações sobre um evento.
2. Adicionar o objeto *Command* em um objeto *Form*, *TextBox*, *List* ou *Canvas*, estas classes são interfaces gráficas da arquitetura J2ME.
3. Adicionar um “receptor” de eventos no objeto que fora adicionado.

Dentro da arquitetura J2ME e do *framework* proposto o que se deve ficar claro é que o *framework* deverá apresentar um casca onde as interfaces que servem de entrada para eventos ou tarefas a serem executadas pelos aplicativos que utilizem a matriz de probabilidades de transição entre tarefas para gerenciar a montagem desta UI.



Figura 5.6 – Calculadora, quando *Command Opções* selecionado.

Para ficar mais claro o uso, apresenta-se abaixo algumas linhas de código para alocar um objeto *Form*, que oferece a opção de exibir vários componentes na tela de um telefone celular e adicionar um objeto *Command* nesse objeto e criar um “receptor” para os eventos detectados.

```
private Form form; // Um formulário
private Command cmd; // Um objeto Command
...
form = new Form("Título"); // Objeto Form
cmd = new Command("Acao", Command.SCREEN, 1); // Objeto Command
...
public void commandAction(Command c, Displayable d) {
    if (c == cmd) {
        // executa uma ação
    }
}
```

Na criação de um novo objeto *Command* para conter informações de evento, existem três parâmetros: o rótulo, o tipo e a prioridade. Apresenta-se uma declaração para um objeto *Command*:

```
cmd = new Command("Sair", Command.EXIT, 1);
```

1. Rótulo: especifica o texto a ser associado ao abjeto *Command*, no exemplo acima seria a palavra “Sair”. O rótulo pode ser mostrado diretamente na tela ou exibido dentro de um *menu*.
2. Tipo: a aplicação usa o tipo do *Command* para especificar a intenção deste *Command*. Por exemplo, se a aplicação especifica que o *Command* é do tipo BACK, e se o dispositivo tem um padrão para inserção deste botão, que representa a ação voltar, na tela. Os tipos definidos são: BACK, CANCEL, EXIT, HELP, ITEM, OK, SCREEN, e STOP.
3. Prioridade: em relação aos outros objetos *Command* que você define, esse valor representa onde esse objeto *Command* cai na escala de prioridades. Quanto mais alto o número, menor a prioridade. Esses valores podem ser úteis para o gerenciador de aplicativos, ao organizar itens que aparecem em um *menu* ou para ordenar botões de software na tela.

O tipo e a prioridade são apenas pedidos do desenvolvedor, o mapeamento de tecla real e a prioridade atribuídas serão decididos e manipuladas pelo dispositivo.

Nota-se então que o desenvolvedor pode enviar dados para que haja o ordenamento dos *Commands*. A intenção do *framework Mandos* é coordenar este ordenamento dos *Commands* com base na matriz de probabilidades de transição entre tarefas.

Com a utilização do *framework Mandos* o *Command*, que representa uma *tarefa* a se executado pelo aplicativo desenvolvido, que apresentar a maior probabilidade de ser executado estará presente no RSK (*right soft key* – tecla direita), enquanto que os outros *Commands*, se houver, estarão em um menu, quando em uma quantidade maior que dois, ordenados de acordo com a probabilidade de mudança de estado associado a cada um.

Neste capítulo refere-se a uma implementação parcial do *framework Mandos*, pois apenas é apresentada a interface responsável pelas *tarefas* que são acionadas pelos *Commands*. Na arquitetura J2ME há outras formas de ativar uma *tarefa*, como por exemplo, em uma *List*, que poderá conter uma lista de opções.

## 5.4 Armazenamento da Matriz de probabilidades de transição entre tarefas

O armazenamento da matriz de probabilidades de transição entre tarefas será realizado em um arquivo no formato XML (eXtensible Markup Language) que é uma recomendação da W3C para gerar linguagens de marcação para necessidades especiais, é capaz de descrever diversos tipos de dados. Seu propósito principal é a facilidade de compartilhamento de informações.

Segundo Bax (2000), um arquivo XML é constituído de elementos, cada elemento possui uma marca inicial (como <titulo> ou <title>), uma marca final (como </titulo> ou </title>), e a informação propriamente dita entre as duas marcas. o formato do arquivo XML, ao invés de descrever como os dados devem ser mostrados, as marcas indicam o que cada dado significa. Qualquer agente (humano ou de software) que receba este documento pode decodificá-lo, e usar os dados como lhe convier.

Um elemento XML pode ser marcado da forma que o autor do documento bem entender, ou seja, com o termo que melhor descreve a informação em sua opinião. Bax (2000), exemplifica que uma livraria poderia usar estes dados para

preencher uma ordem de compra; um analista de mercado para descobrir quais livros são mais populares; um indivíduo poderia armazená-lo num banco de dados como um registro de suas compras, etc.

O formato do arquivo para armazenamento da matriz de probabilidades de transição de tarefas XML adotado fora o XML, com já declarado, por ser simples de entendimento para o ser humano, pois o propósito é justamente obter uma forma fácil e viável de alteração no aplicativo.

Para acessar as informações armazenadas nos arquivos XMLs utilizando a arquitetura J2ME, utiliza-se a JSR 172, que é uma API padrão para serviços Web, dentre os quais apresenta métodos para realizar o parse de arquivos XMLs. A JSR 172 implementa um subconjunto da JAXP 1.2 (Java API for XML Processing). Essencialmente isto determina que a especificação SAX 2.0 (Simple API for XML) deve ser suportada. (Sony Ericsson, 2006)

A seguir apresenta-se o formato do arquivo utilizado pelo protótipo desenvolvido.

```
<xml>
  <tarefas>
    < tarefa nome="Configurar hora" id="268506248" probabilidades="0.093;0.119;0.093;0.093;0.103;0.393;0.103"
checkbox="nao"/>
    < tarefa nome="Alarme ligado" id="268506240" probabilidades="0.291;0.130;0.192;0.096;0.096;0.096;0.096"
checkbox="sim"/>
    < tarefa nome="Config. dias" id="268436469" probabilidades="0.099;0.208;0.099;0.297;0.099;0.099;0.099"
checkbox="nao"/>
    < tarefa nome="Definir melodias" id="268442695" probabilidades="0.400;0.100;0.100;0.100;0.100;0.100;0.100"
checkbox="nao"/>
    < tarefa nome="Relogio" id="268435948" probabilidades="0.097;0.097;0.107;0.097;0.115;0.389;0.097"
checkbox="nao"/>
    < tarefa nome="Pausa autom." id="268442667" probabilidades="0.100;0.100;0.100;0.100;0.100;0.100;0.400"
checkbox="sim"/>
    < tarefa nome="Sair" id="268436111" probabilidades="0.000;0.000;0.000;0.000;0.000;0.000;1.000" checkbox="nao"/>
  </tarefas>
  <geral>
    <framework nivel="3" contador_observacoes="100"/>
  </geral>
</xml>
```

Dentro da sessão xml encontram-se as sessões *tarefas* e *geral*.

A sessão *tarefas* é composta por diversos elementos de nome tarefa, que representam o conceito *tarefa* apresentado neste trabalho.

O *elemento tarefa* contém os seguintes atributos:

- *nome* – nome que representa a tarefa, ela será mostrada como rótulo do *Command* quando o id não for informado, se o id for informado, este nome servirá somente para a identificação da tarefa apenas pela leitura do arquivo XML. Este é um atributo obrigatório.

Exemplo: *nome="Configurar hora"*. Identifica neste elemento contém os dados relativos à tarefa “Configurar hora”.

- *id* – É um número inteiro usado pela JSR 238 (JCP, 2005) para o processo de internacionalização, que é um processos de desenvolvimento e/ou adaptação de um produto, em geral softwares de computadores. Este número é correspondente ao rótulo a ser mostrado no *Command*, assim o rótulo será mostrado de acordo com a linguagem corrente do aparelho. A internacionalização é importante porque permite que o desenvolvedor de software respeite as particularidades de cada língua e cultura de cada país. Este é um atributo opcional, se o mesmo não existir o nome a ser mostrado no rótulo do *Command* será o contido no atributo nome. A Figura 5.7 ilustra um aplicativo internacionalizado, note que o rótulo alterou com a mudança do idioma, do lado esquerdo, o inglês, do lado direito, o alemão.



Figura 5.7: Calculadora, quando em idioma inglês e alemão, respectivamente.

Exemplo: `id="268436111"`. Este ID equivale a string `Sair`, em português, ou seja, o `Command` que representa a tarefa `sair` receberá a string internacionalizada, ou seja, receberá a palavra `Sair` de acordo com o corrente linguagem do telefone móvel.

- *probabilidades* – contém as probabilidades de transição entre os estados, a seqüência das probabilidades está sempre associada com a ordem das tarefas, ou seja, a primeira probabilidade da lista se refere à primeira tarefa, por exemplo, no arquivo descrito acima a primeira probabilidade está associada a ela mesma, a próximo se associa com a segunda tarefa descrita e assim por diante, até a última probabilidade da lista está associada ao estado final. Se o nível 3 estiver sendo utilizado, estas probabilidades serão alteradas, de acordo com o uso do usuário, tais valores serão agora freqüências relativas, como descrito no item 4.1.2.2. O novo valor da probabilidade será calculado pela seguinte fórmula:

$$P_{n \text{ novo}} = ((P_{n \text{ antigo}} * n) + 1) / (n + 1)$$

, onde:

$P_{n \text{ novo}}$  – nova freqüência relativa de transição para uma tarefa  $n$ .

$P_{n \text{ antigo}}$  – anterior freqüência relativa de transição para uma tarefa  $n$ .

$n$  – número total de observações, que será armazenado no atributo *contador\_observacoes*.

Este é um atributo obrigatório.

Exemplo: `probabilidades="0.093;0.119;0.093;0.093;0.103;0.393;0.103"`, A primeira tarefa será o primeiro elemento da seção tarefas e assim por diante, seguindo a ordem, a lista de probabilidades, também segue a ordem apresentada, ou seja, a primeira probabilidade, no caso, 0.093, se refere a primeira tarefa, o que significa que a tarefa que contém como atributo a lista de probabilidade descrita tem a probabilidade de 0.093 de a partir do corrente estado mudar para a primeira tarefa, assim como tem 0.119 de mudar para segunda tarefa, assim por diante.

Para melhor ilustração deste atributo, recorre-se ao XML apresentado, ao consultar o elemento que representa a tarefa “Alarme ligado”, pode-se concluir que o mesmo possui uma probabilidade de 0.291 de efetuar agora a tarefa “Configurar hora”, e de uma probabilidade de 0.13 de efetuar a mesma tarefa novamente (“Alarme ligado”), assim como tem uma probabilidade de 0.192 de efetuar a tarefa “Config. dias” e uma probabilidade de 0.096 de “Definir melodia” e uma probabilidade de 0.096 de efetuar a tarefa “Relógio” e uma probabilidade de 0.096 de efetuar a tarefa “Pausa autom.” e, finalmente, a probabilidade de 0.096 de sair do aplicativo.

- *checkbox* – Indica se um checkbox deve ser mostrado ao lado do rótulo do Command. O checkbox é um indicador localizado ao lado do rótulo do Command que indica o estado da tarefa, quando ativado, um check é mostrado dentro do quadrado, como ilustrado na Figura 5.8. Este é um atributo opcional, se este atributo não constar em um elemento que representa uma tarefa o checkbox não será mostrado.



Figura 5.8: Lista com alguns *Commands* com checkbox (“Alarme ligado” e “Pausa autom,”) indicando que a tarefa está no estado ativo.

A sessão *geral* contém algumas informações sobre o uso do framework Mandos.

- *nível* – indica o nível de utilização do *framework*, pode assumir três valores, como descrito abaixo:
  1. “1” – indica que o framework Mandos será utilizado no nível 1, no modo de inicialização estático, e se comportará como descrito na seção 4.2 deste trabalho;
  2. “2” – indica que o framework Mandos será utilizado no nível 2, no modo de inicialização com transição dinâmica das tarefas, e se comportará como descrito na seção 4.2 deste trabalho;
  3. “3” – indica que o framework Mandos será utilizado no nível 3, no modo de adaptação da matriz de transição, e se comportará como descrito na seção 4.2 deste trabalho;
 Este é um atributo obrigatório.
- *contador\_observacoes* – Este atributo é utilizado pelo *framework Mandos* apenas quando o nível 3 do *framework* está habilitado. O mesmo se refere à quantidade de vezes de observação do evento seleção de uma tarefa, a cada escolha de uma tarefa do aplicativo que utilizado o *framework*

*Mandos no nível 3* este atributo é atualizado. Isto é necessário para a realização do cálculo da frequência relativa, citada nesta seção e na seção 4.1.2.2.1 deste trabalho.

## 5.5 Utilizando o Framework Mandos

Apresentou-se a maneira corrente de utilizar a arquitetura J2ME como relação aos *Commands*, sendo que pode haver variações da maneira apresentada, abaixo se apresenta como seria a utilização do *framework Mandos*, procurou-se desenvolver uma forma onde a adaptação de código escrito sem utilizar o *framework* seja rápida e simples.

```
// Um formulário
private Form form;
// Um conjunto de Commands das tarefas
static Command[] COMMANDS_OPTIONS = new Command[] { TASK_01,
                                                    TASK_02, TASK_03};
// Indica o estado corrente, deve ser atualizado de acordo com o acionamento de uma nova tarefa
private int currentTask;

...

// Este método lê o XML que contém a matriz de transição e informações sobre o uso do framework, //
isto é necessário para não deixar o aplicativo lendo, pois a leitura de um arquivo é lenta.
// Recomenda-se utilizar este método no construtor. Com a leitura feita, todos os dados são
// carregados para serem utilizados de maneira rápida.
ActionsPool.fillPool("0:/misc/exemplo.xml"); // 0:/misc/exemplo.xml é a localização do XML no
// celular. Recomenda-se que o arquivo XML esteja em um lugar seguro, sem o acesso do usuário.
// Para que não seja corrompido.

...
form = new Form("Título"); // Objeto Form

public void commandAction(Command c, Displayable d) {
    if (c == CMD_OPTIONS) {
        // mostra o menu de opções ordenado
        Displayable alarmOptionsMenu = UsabilityUI.showOptions(
            "0:/misc/despertador.txt",
            currentTask, COMMANDS_OPTIONS,
            this, alarmMainScreen, checked);

        display.setCurrent(alarmOptionsMenu);
    }

    else if (c == CMD_TAREFA_1) {
        // colhe dados do usuário e atualiza o arquivo XML com os dados, se o nível 3 estiver
        // habilitado
        ActionStore.updateActions(currentAction, 0, "0:/misc/despertador.txt");
        //atualiza a tarefa corrente
        currentTask = 0;
    }
}
```

### 5.5.1 Métodos do Framework Mandos para tratar os Commands

- `void fillPool(String xmlPath);`  
Este método é utilizado para evitar múltiplos e desnecessários acessos ao arquivo XML, o que poderia deixar o aplicativo lento, então se aconselha que use este método apenas uma vez antes de mostrar o *menu de opções* ao usuário. O mesmo tem o intuito de ler os dados do XML e armazená-lo na memória destinada ao aplicativo na RAM do celular, fazendo com que o acesso a estes dados seja mais rápido. O parâmetro de entrada é uma string com o endereço de localização do XML no aparelho celular.
- `Displayable showOptions( String pathXML,  
int currentTask, Command[] commands,  
CommandListener commandListener,  
Displayable pDisplayable, boolean[] checkBox);`  
Este método é responsável pela montagem da *user interface* como o *framework* se propõe a fazer, ou seja, baseado no arquivo XML, os *Commands* serão ordenado e adicionados no *Displayable* a ser exibido, que será retornado pelo método. Os parâmetros de entradas são: o endereço do XML no celular, a atual tarefa, um vetor com todos os *Commands* a serem ordenados, uma instância da classe *CommandListener*, para tratar as ações do *Commands*, uma instância da classe *Displayable* com a tela a ser mostrado quando a lista com os *Commands* for cancelada e um vetor com os valores dos *checkbox*, se houver, senão pode ser passado `null`.
- `void updateActions(int currentTask, int lastTask, String xmlPath);`  
É utilizado no nível 3 do *framework* para atualizar os valores da matriz de transição de tarefa. Como parâmetro de entrada tem-se: a corrente tarefa, a tarefa anterior e o endereço do arquivo XML no celular.

# Capítulo 6

## Descrição do Experimento

Este capítulo descreve o experimento realizado para investigar qual a influência exercida pelo uso do Método de interação Mandos na usabilidade de um aplicativo embarcado em um telefone móvel.

A seguir serão descritas todas as etapas para a montagem e realização destes ensaios de interação segundo o Método de Medida da Performance (apresentado no item 3.2). Todos os modelos dos documentos utilizados para a realização do experimento estão em anexo (Anexo A ao Anexo H).

### 6.1 Definição do produto a ser avaliado

Serão dois aplicativos a serem avaliados neste experimento, a calculadora e o despertador (ambos apresentados no item 3.7.8).

Haverá duas versões de cada aplicativo, sendo que uma será desenvolvida com o uso do *framework Mandos nível 3*, contendo todas as funcionalidades da versão original, que não sofrerá qualquer modificação de como está instalado no aparelho BenQ-Siemenes SL75 com data de produção de 5 de março de 2007 e com versão do software de número 39.

### 6.2 Definição do contexto de uso

Com intuito de dar ênfase na avaliação no *método de interação Mandos*, as aplicações que foram avaliadas podem ser utilizadas por qualquer usuário que possua um telefone celular, sendo todos os usuários considerados um cliente em potencial.

### 6.3 Definição dos objetivos e contextos de avaliação

O objetivo do experimento é avaliar se o uso do *método de interação Mandos* resulta em uma melhoria na usabilidade de um aplicativo.

### 6.3.1 Definição da amostra de participantes

Apesar do teste a ser realizado neste trabalho ter como intuito verificar o grau de usabilidade de um aplicativo, e assim poder verificar se houve um acréscimo ou não do grau de usabilidade com o uso do método de interação Mandos, escolheu-se utilizar um número maior de participantes sugerido no item 3.6 deste trabalho, tentando assim abranger uma gama maior do público-alvo e principalmente ter dados suficientes para comparar estatisticamente e assim mensurar o grau de variação na usabilidade com o uso do método de interação proposto, não apenas descobrindo problemas de usabilidade.

O número de participantes determinado nos ensaios foi de 20 pessoas. se caso haja a necessidade de analisar os resultados estatisticamente, segundo Rubin (1994) e Spyridakis (1992) 10 pessoas é o número mínimo de participantes por grupo de teste.

Foram selecionados 10 homens e 10 mulheres. Cada um dos usuários participou de um único ensaio de interação com cada versão dos aplicativos.

A escolha dos usuários deve sempre ser realizada de forma que eles sejam tão representativos quanto possível em relação aos usuários reais do sistema.

Como o escopo em relação ao usuário não foi limitado a um público-alvo, com base na pesquisa de Betiol (2002), para melhor atender o pressuposto, definiu-se que nesta pesquisa os participantes deveriam atender aos seguintes critérios:

- Não são usuários do modelo de telefone celular utilizado nos testes. Como recomendado por Nielsen (2000) isto evita que haja diferentes níveis de experiência em relação ao manuseio do aparelho.
- Usuários entre 16 e 40 anos de idade. Esta faixa etária foi a escolhida pois ela representa a maior concentração de usuários de telefone móvel no Brasil atualmente, correspondendo a mais de 65% do total (ANATEL, 2003).
- Os usuários deveriam ter diferentes formações e ocupações. Como constatado por Rocha (2003) “não é conveniente testar uma interface voltada para o público em geral e utilizar estudantes de computação como grupo de teste: eles certamente não são representativos da população de usuários alvo”,
- A Tabela 6.1 apresenta a distribuição dos participantes de acordo com a faixa etária e o sexo.

Tabela 6.1: Distribuição dos participantes por faixa etária e sexo

Faixa etária	Sexo		Total
	masculino	feminino	
16 – 20	1	3	4
21 – 25	3	4	7
26 – 30	4	2	6
30 – 40	2	1	3
Total	10	10	<b>20</b>

### 6.3.2 Definição da lista das tarefas

A elaboração das tarefas foi baseada no fato de simular o comportamento do usuário ao longo do tempo com o uso do telefone móvel, de maneira que o aplicativo avaliado vá se adaptando ao uso e preferências demonstradas na lista de tarefas.

As tarefas tentam representar as seqüências de execução mais comuns de serem executadas nos aplicativos avaliados.

Segue a lista de tarefas para o teste de usabilidade utilizada:

No aplicativo *despertador*:

- Iniciar o despertador, verificar o status do despertador e sair;
- Iniciar o despertador, ativar o alarme e sair;
- Iniciar o despertador, desativar o alarme e sair;
- Iniciar o despertador, ativar o alarme, configurar hora (8:00 AM) e sair;
- Iniciar o despertador, ativar o alarme, configurar hora (7:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta e sexta) e sair;
- Iniciar o despertador, configurar hora (6:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta, sexta e sábado), definir melodia e sair;
- Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a mais) e sair;
- Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a menos), habilitar pausa automática e sair;

No aplicativo *calculadora*:

- Iniciar a calculadora, somar 42 com 33, ver o resultado e sair;
- Iniciar a calculadora, multiplicar 56 com 22, ver o resultado, salvar o resultado na memória, somar 47 com 11, somar o resultado obtido com o armazenado na memória e sair;
- Iniciar a calculadora, abrir a memória, limpar a memória, somar 30 com 65 e entrar no modo conversor, voltar para calculadora e sair;
- Iniciar a calculadora, ir para o modo ampliado, tirar a raiz quadrada de 45, salvar na memória, ir para o modo básico, diminuir 32 de 14,5 e somar com o resultado armazenado na memória e sair;
- Iniciar a calculadora, realizar a operação 20 dividido por 45, ver o resultado, limpar a tela e sair;

Todas as tarefas foram realizadas no aplicativo com e sem o uso do *método de interação Mandos nível 3* para que o usuário possa avaliar a diferença entres ambos métodos de interação.

### 6.3.3 Escolha das medidas e métricas de usabilidade

A definição de usabilidade adotada neste trabalho foi a de Nielsen (1998).

Através da avaliação de cada um dos quesitos que compõem a usabilidade, segundo a definição adotada, será avaliado o nível de aprimoramento na usabilidade em aplicativos embarcados em telefones móveis que utilizam o *método de interação Mandos*.

Os itens serão avaliados pelos participantes que realizarão os ensaios de interação usando a escala Likert (o uso desta escala é recomendado para pesquisas com questionários) para cada item, apresentados a seguir, que compõem a usabilidade. O texto que define o item foi o mesmo apresentado ao participante.

- Intuitividade - Entre os atributos que compõe a usabilidade, o mais importante é a intuitividade. Para que ocorra minimamente uma interação, a interface deve apresentar características que facilitem sua utilização permitindo que usuários básicos ou avançados possam aprender seus recursos de forma clara e objetiva.
- Eficiência - A eficiência é determinada através de quanto tempo é gasto utilização do software para realização de uma tarefa. Quanto menor o tempo gasto, ou menor o esforço despendido, mais eficiente é o aplicativo.
- Memorização - Denota o quanto o aplicativo é fácil de ser memorizado, ou seja, após a aprendizagem de uma tarefa, se o aplicativo auxilia de forma a tornar a realização desta tarefa mais fácil de ser memorizada.
- Redução de Erros - É considerado erro qualquer ação que leve o usuário a não executar determinada tarefa não se levando em conta os possíveis impactos causados por diferentes tipos de erros. Um bom aplicativo apresenta uma interação com o usuário de forma que o mesmo tenha menor chance de cometer erros em uma tarefa.
- Satisfação - A satisfação representa o quão agradável deve ser a interação do usuário com o sistema.

A nota será dada para uma versão do aplicativo que utiliza o *método de interação Mandos* e para o mesmo aplicativo, só que em uma versão que não utiliza o método citado.

## 6.4 Condução dos ensaios de interação

As avaliações foram realizadas ao longo de uma semana, sendo que o autor deste trabalho foi o único avaliador responsável pela condução de todos os ensaios de interação.

Cada um dos ensaios de interação, com as duas aplicações, com e sem o uso do *método de interação Mandos*, durou em média quarenta minutos. Todos os testes seguiram rigorosamente o mesmo roteiro que foi elaborado a partir de (Ferreira, 2004; Betiol, 2002; Rubin, 1994; Dumas, 1999; Nielsen, 2003). Este roteiro está descrito a seguir.

### 6.4.1 Introdução do participante ao ambiente do teste

O participante foi recebido e lhe foi explicado como seria conduzida a avaliação, o que ele teria que fazer e quanto tempo duraria a sessão. Segundo Betiol (2002) é muito importante que o avaliador forneça ao participante todas as informações sobre a condução da avaliação para que este possa se sentir à vontade. Na literatura (Nielsen, 1993; Dumas, 1999; Rubin, 1994) é fortemente recomendado que o avaliador explique claramente ao participante que não é ele quem está sendo avaliado, e sim que o objetivo da avaliação é a aplicação, no caso o *método de interação Mandos*. Com a orientação do trabalho desenvolvido por Betiol (2002), durante a realização destes ensaios de interação em nenhum momento foi

mencionada ao usuário a palavra “teste”. No recrutamento os participantes foram convidados a participar de uma “pesquisa” de usabilidade e não de um “teste” de usabilidade.

Foram mostrados aos participantes todos os equipamentos utilizados no teste, e também foi explicado que seu rosto não seria fotografado ou filmado e que todo o procedimento seria anônimo.

Foi pedido ao participante que preenchesse um rápido questionário com alguns dados sobre o seu perfil. Um modelo de questionário para registrar o perfil do usuário pode ser encontrado no Anexo C.

### 6.4.2 Treinamento do participante

Após completar o questionário, o participante recebe o Script de Orientação do teste, que pode ser encontrado no Anexo D.

O avaliador leu o script junto com o participante reforçando que o centro da avaliação é o produto e não o participante em si. O participante foi informado que ele estará sendo observado e fotografado e que a integridade do participante será totalmente resguardada, sendo utilizada a observação e as imagens somente para fins de análise do teste. O avaliador deve reforçar outras informações constantes do script e retirar dúvidas do participante sobre a sessão de teste.

O participante foi introduzido ao *método de interação Mandos* observando a diferença entre o método utilizado no telefone celular, para estar ciente do principal fator a ser avaliado.

Após serem passadas as orientações, o avaliador informou ao participante que ele pode utilizar o sistema livremente durante cinco minutos.

### 6.4.3 Execução das tarefas

Após o treinamento do participante, uma lista de tarefas foi entregue ao participante (a citada lista pode ser encontrada no Anexo E deste trabalho).

Os acontecimentos observados pelo avaliador deverão ser registrados no formulário de Coleta de Dados pelo Avaliado, que se encontra no Anexo F deste trabalho.

### 6.4.4 Questionários e discussões

Depois de completadas todas as tarefas, foi entregue ao participante o Questionário de Avaliação do Sistema pelo Participante para ser preenchido. Este questionário pode ser encontrado no Anexo G deste trabalho.

Ao final da avaliação foi realizada uma discussão com o participante para obter comentários gerais sobre a avaliação e esclarecimentos sobre as possíveis dificuldades que ele poderia ter enfrentado durante a interação. No Anexo H, encontra-se um guia para a realização da citada discussão, chamado de Tópicos para Questionamento.

Após a discussão a avaliação estava encerrada.

## 6.5 Análise estatística

Após a realização de todos os ensaios de interação foram quantificados os dados dos questionários aplicados. Com os dados quantificados foi realizado testes estatísticos para verificar se realmente houve um aprimoramento na usabilidade de um aplicativo projeto para telefones móveis que utiliza o *método de interação Mandos*.

Segundo Wainer (2007) testes estatísticos são procedimentos que fazem uma particular pressuposição sobre os dados, a chamada hipótese nula, e mais uma série de outras pressuposições (condições de teste) sobre os dados e calculam a probabilidade que alguma propriedade relacionada aos dados seja verdadeira, dadas as pressuposições. Essa probabilidade calculada é chamada valor- $p$  ou  $p$ -value. Se as condições do teste são verdadeiras e o valor- $p$  é suficientemente baixo, então o pesquisador pode assumir que a hipótese nula é falsa. Em relação ao valor- $p$ , podem ser considerados os seguintes valores:

- valor- $p < 0,01$  = elevada significância estatística, é uma evidência muito forte contra a hipótese nula.
- $0,01 < \text{valor-}p < 0,05$  = estatisticamente significativa, é uma evidência moderada contra a hipótese nula.
- $0,05 < \text{valor-}p < 0,1$  = estatisticamente significativa, é uma evidência fraca contra a hipótese nula.
- valor- $p > 0,1$  = evidência insuficiente contra a hipótese nula, ela não deve ser rejeitada.

Além da hipótese nula, adotou-se uma hipótese alternativa que irá ser confirmada com o valor- $p$ , que a média obtida por cada item da usabilidade avaliado do *método de interação Mandos* seja superior a obtida pelo método normal de interação.

Para a realização do cálculo do valor- $p$ , inicialmente pensou-se em utilizar o teste T pareado, pois o mesmo é usado quando se for verificar se a média de um conjunto de medidas é maior que a média de outro conjunto e quando cada medida de um conjunto pode se colocada em correspondência com uma de outro conjunto, mas para utilização deste teste é necessário obedecer as seguintes condições:

- As variâncias dos dois conjuntos são iguais
- Que os dados dos dois conjuntos estão distribuídos segundo uma normal

Para verificar se os dados dos conjuntos estavam distribuídos dentro de uma normal, utilizou-se o teste de Shapiro-Wilk. (Ara, 2003) O teste F (Winter, 2002) é utilizado para verificar se a variabilidade dos dados é decorrente do fato de que os mesmos pertencem a diferentes contextos de avaliação ou se é devido ao acaso (variabilidade residual).

O teste de Shapiro-Wilk atestou que alguns conjuntos não estavam distribuídos dentro de uma normal, o que impossibilitou o uso do teste T pareado.

Então para a determinação do valor- $p$  foi utilizado o teste de Wilcoxon signed-rank test, que segundo Wainer (2007) deve ser usado se as condições do teste T não são verdadeiras (não normalidade no caso). Este é um teste não-paramétrico, o que significa que entre suas condições de teste não há qualquer pressuposição que os dados têm alguma distribuição fixa, por este motivo são considerados mais fracos que seus correspondentes paramétricos.

Para realização dos cálculos estatísticos foi utilizado o software Analyse-it. (Analyse-it, 2007)

Na apresentação dos resultados, além do resultado do valor- $p$ , para melhor visualização dos dados será apresentado também um diagrama de caixa, que segundo Montgomey (2004) é uma apresentação gráfica que descreve simultaneamente várias características importantes de um conjunto de dados, tais como: centro, dispersão, desvio de simetria e identificação das observações que estão, não geralmente, longe do seio dos dados.

# Capítulo 7

## Resultados

Apresenta-se neste capítulo os resultados obtidos no experimento realizado.

Primeiramente, os dados estão dispostos de acordo com o quesito que compõe a usabilidade: intuitividade, eficiência, erros, memorização e satisfação, após é feita uma análise geral sobre a usabilidade e sobre a percepção geral do participante do experimento sobre o método de interação Mandos, por fim é feita uma listagem das limitações do experimento que poderá acarretar alguma distorção no resultado obtido.

### 7.1 Resultados do quesito Intuitividade

A Tabela 7.1 contém uma coletânea de informações sobre os participantes da pesquisa associado com as notas dadas para o quesito intuitividade para os aplicativos utilizados que faziam o uso do *método de interação Mandos* (coluna “com o uso”) e os que não faziam uso (coluna “sem o uso”).

Os dados resultantes da análise estatística, relativos ao requisito intuitividade, estão apresentados na Tabela 7.2.

O valor- $p$  obtido através do teste de comparação de média Wilcoxon signed-rank foi de 0,0028, o que significa que há uma evidência estatística muito elevada de que a média do quesito intuitividade obtida pelos aplicativos que faziam uso do *método de interação Mandos* é maior do que as dos aplicativos que não utilizaram o citado método.

A Figura 7.1 ilustra, através do diagrama de caixa (95% CI Notched Skeletal Boxplot) e de diamante (95% CI Mean Diamond), os dados obtidos de maneira estatística.

Através das análises estatísticas dos dados obtidos pelo experimento descrito no capítulo 6, pode-se assumir que a Hipótese I, descrita na seção 4.4 deste trabalho, de que a avaliação dos aplicativos que não utilizam o *método de interação Mandos*, no que se diz respeito à intuitividade, será melhor do que a dos aplicativos que utilizam o citado método de interação é falsa, podendo –se afirmar que a avaliação dos aplicativos que utilizam o *método de interação Mandos* no que se diz respeito à intuitividade é melhor do que a dos aplicativos que não utilizam o *método de interação Mandos*, o que acarreta em uma melhoria parcial na usabilidade,

confirmando parcialmente que é método *de interação Mandos* apresenta uma melhoria de usabilidade nos aplicativos que o utilizam.

Tabela 7.1: Notas atribuídas ao quesito Intuitividade para os aplicativos que utilizam e os que não utilizam o *Método de interação Mandos*

Dados dos Participantes						Intuitividade	
idade	sexo	Escolaridade	profissão	tempo de uso de app.	horas de uso por semana	sem o uso	com o uso
16	fem.	ens. médio incompleto	Estudante (eletrotécnica)	3 a 4 anos	2 a 5 horas	3	4
18	fem.	ens. médio completo	Aux. Administrativo	2 a 3 anos	menos de 2 horas	4	4
19	fem.	superior incompleto	Estudante (direito)	3 a 4 anos	2 a 5 horas	2	4
20	masc.	superior incompleto	Desenvolvedor de SW	1 a 2 anos	menos de 2 horas	4	3
21	fem.	superior incompleto	Estudante (medicina)	mais de 4 anos	menos de 2 horas	2	4
21	fem.	superior incompleto	Estudante (letras)	mais de 4 anos	menos de 2 horas	5	2
23	fem.	superior incompleto	Servidora pública	mais de 4 anos	menos de 2 horas	2	4
24	fem.	superior completo	Eng. de Qualidade	mais de 4 anos	menos de 2 horas	3	5
24	masc.	superior incompleto	Analista de Produção	3 a 4 anos	menos de 2 horas	3	4
25	masc.	superior completo	Analista de Sistemas	2 a 3 anos	5 a 10 horas	3	4
25	masc.	superior completo	Eng. de SW	mais de 4 anos	5 a 10 horas	3	4
26	masc.	superior incompleto	Desenvolvedor de SW	mais de 4 anos	2 a 5 horas	3	5
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	4
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	4
28	masc.	superior incompleto	Desenvolvedor de SW	3 a 4 anos	2 a 5 horas	3	4
29	fem.	superior completo	Consultora de Vendas	mais de 4 anos	menos de 2 horas	4	5
30	fem.	superior completo	Eng. de processo	3 a 4 anos	2 a 5 horas	3	3
33	fem.	superior completo	Tec. Judiciária	3 a 4 anos	menos de 2 horas	3	5
36	masc.	superior incompleto	Assis. Administrativo	2 a 3 anos	menos de 2 horas	4	5
36	masc.	superior incompleto	Representante comercial	3 a 4 anos	menos de 2 horas	2	4

Tabela 7.2: Dados estatísticos obtidos para o quesito Intuitividade

<b>Intuitividade</b>			
<b>Método de interação</b>	<b>número de amostras</b>	<b>média</b>	<b>desvio-padrão</b>
<b>Mandos</b>	20	4,1	0,8
<b>Normal</b>	20	3,1	0,8
<b>Mandos – Normal</b>	20	1	1,2
<b>Diferença entre pares</b>	<b>n</b>	<b>soma rank</b>	<b>média rank</b>
<b>Positiva</b>	16	145,5	9,22
<b>Negativa</b>	2	23,5	11,75
<b>Zero</b>	2		
<b>valor-p (Wilcoxon signed-rank test)</b>		<b>0,0028</b>	

O *método de interação Mandos* acabou mostrando-se de fácil aprendizagem e intuito, contrariando a expectativa inicial. Acredita-se que o principal motivo para este comportamento é que nenhum dos participantes tenha tido contato anterior com os aplicativos, nem o telefone móvel avaliado, o que significa que nenhum dos participantes estava condicionado com o uso dos aplicativos. O condicionamento do usuário de um aplicativo foi o que levou a hipótese de uma avaliação negativa do *método de interação Mandos*, que foi rejeitada no experimento realizado.

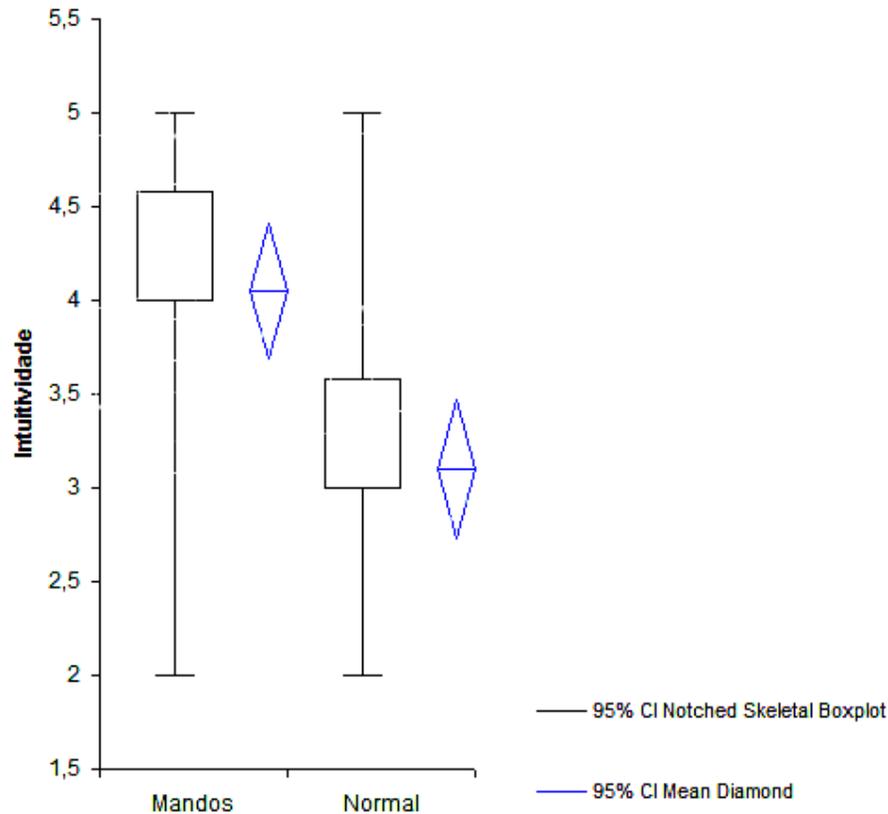


Figura 7.1: Diagrama de Caixa e de Diamante dos valores do quesito Intuitividade

## 7.2 Resultados do quesito Eficiência

A Tabela 7.3 contém uma coletânea de informações sobre os participantes da pesquisa associado com as notas dadas para o quesito eficiência para os aplicativos utilizados que faziam o uso do *método de interação Mandos* (coluna “com o uso”) e os que não faziam uso (coluna “sem o uso”).

Os dados resultantes da análise estatística, relativos ao requisito eficiência, estão apresentados na Tabela 7.4.

A Figura 7.2 ilustra, através do diagrama de caixa (95% CI Notched Skeletal Boxplot) e de diamante (95% CI Mean Diamond), os dados obtidos de maneira estatística.

Tabela 7.3: Notas atribuídas ao quesito Eficiência para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos

Dados dos Participantes						Eficiência	
idade	sexo	escolaridade	Profissão	tempo de uso de app.	horas de uso por semana	sem o uso	com o uso
16	fem.	ens. médio incompleto	Estudante (eletrotécnica)	3 a 4 anos	2 a 5 horas	2	5
18	fem.	ens. médio completo	Aux. Administrativo	2 a 3 anos	menos de 2 horas	4	5
19	fem.	superior incompleto	Estudante (direito)	3 a 4 anos	2 a 5 horas	2	4
20	masc.	superior incompleto	Desenvolvedor de SW	1 a 2 anos	menos de 2 horas	3	5
21	fem.	superior incompleto	Estudante (medicina)	mais de 4 anos	menos de 2 horas	1	3
21	fem.	superior incompleto	Estudante (letras)	mais de 4 anos	menos de 2 horas	5	4
23	fem.	superior incompleto	Servidora pública	mais de 4 anos	menos de 2 horas	2	4
24	fem.	superior completo	Eng. de Qualidade	mais de 4 anos	menos de 2 horas	2	5
24	masc.	superior incompleto	Analista de Produção	3 a 4 anos	menos de 2 horas	2	4
25	masc.	superior completo	Analista de Sistemas	2 a 3 anos	5 a 10 horas	4	5
25	masc.	superior completo	Eng. de SW	mais de 4 anos	5 a 10 horas	3	5
26	masc.	superior incompleto	Desenvolvedor de SW	mais de 4 anos	2 a 5 horas	2	4
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	5
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	4	4
28	masc.	superior incompleto	Desenvolvedor de SW	3 a 4 anos	2 a 5 horas	2	4
29	fem.	superior completo	Consultora de Vendas	mais de 4 anos	menos de 2 horas	3	4
30	fem.	superior completo	Eng. de processo	3 a 4 anos	2 a 5 horas	2	4
33	fem.	superior completo	Tec. Judiciária	3 a 4 anos	menos de 2 horas	2	5
36	masc.	superior incompleto	Assis. Administrativo	2 a 3 anos	menos de 2 horas	4	5
36	masc.	superior incompleto	Representante comercial	3 a 4 anos	menos de 2 horas	2	4

Tabela 7.4: Dados estatísticos obtidos para o quesito Eficiência

<i>Eficiência</i>			
<i>Método de interação</i>	<i>número de amostras</i>	<i>média</i>	<i>desvio-padrão</i>
<b>Mandos</b>	20	4,4	0,6
<b>Normal</b>	20	2,7	1,0
<b>Mandos – Normal</b>	20	1,7	1,0
<i>Diferença entre pares</i>	<i>n</i>	<i>soma rank</i>	<i>média rank</i>
<b>Positiva</b>	18	187,0	10,39
<b>Negativa</b>	1	3,0	3,00
<b>Zero</b>	1		
<b>valor-p (Wilcoxon signed-rank test)</b>		<b>&lt;0,0001</b>	

O valor-p obtido através do teste de comparação de média Wilcoxon signed-rank foi menor do que 0,0001, o que significa que há uma evidência estatística muito

elevada de que a média do quesito eficiência obtida pelos aplicativos que faziam uso do *método de interação Mandos* é maior do que as dos aplicativos que não utilizaram o citado método.

Através das análises estatísticas dos dados obtidos pelo experimento descrito no capítulo 6, pode-se assumir que a Hipótese II, descrita na seção 4.4 deste trabalho, de que a avaliação dos aplicativos que utilizam o *método de interação Mandos*, no que se diz respeito à eficiência, será melhor do que a dos aplicativos que não utilizam o citado método de interação, é verdadeira, o que acarreta em uma melhoria parcial na usabilidade, confirmando parcialmente que o método *de interação Mandos* apresenta uma melhoria de usabilidade nos aplicativos que o utilizam.

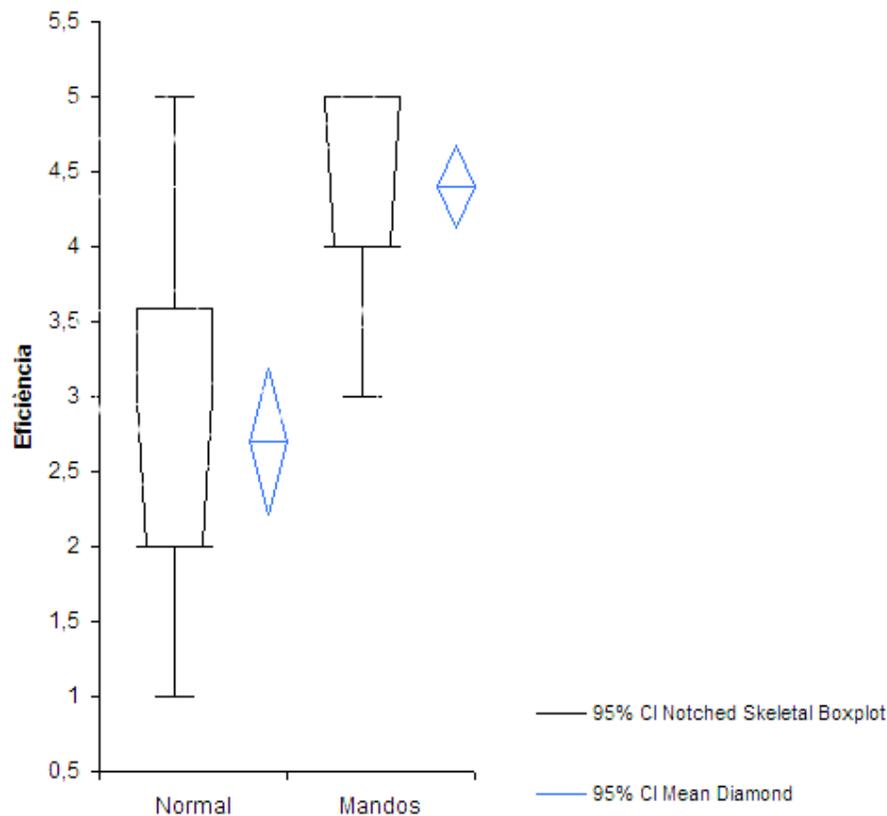


Figura 7.2: Diagrama de Caixa e de Diamante dos valores do quesito Eficiência

### 7.3 Resultado do quesito Memorização

A Tabela 7.5 contém uma coletânea de informações sobre os participantes da pesquisa associado com as notas dadas para o quesito memorização para os aplicativos utilizados que faziam o uso do *método de interação Mandos* (coluna “com o uso”) e os que não faziam uso (coluna “sem o uso”).

Os dados resultantes da análise estatística, relativos ao requisito memorização, estão apresentados na Tabela 7.6.

Tabela 7.5: Notas atribuídas ao quesito Memorização para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos

Dados dos Participantes						memorização	
idade	sexo	escolaridade	profissão	tempo de uso de app.	horas de uso por semana	sem o uso	com o uso
16	fem.	ens. médio incompleto	Estudante (eletrotécnica)	3 a 4 anos	2 a 5 horas	4	4
18	fem.	ens. médio completo	Aux. Administrativo	2 a 3 anos	menos de 2 horas	4	3
19	fem.	superior incompleto	Estudante (direito)	3 a 4 anos	2 a 5 horas	4	2
20	masc.	superior incompleto	Desenvolvedor de SW	1 a 2 anos	menos de 2 horas	4	4
21	fem.	superior incompleto	Estudante (medicina)	mais de 4 anos	menos de 2 horas	4	4
21	fem.	superior incompleto	Estudante (letras)	mais de 4 anos	menos de 2 horas	4	3
23	fem.	superior incompleto	Servidora pública	mais de 4 anos	menos de 2 horas	4	3
24	fem.	superior completo	Eng. de Qualidade	mais de 4 anos	menos de 2 horas	4	5
24	masc.	superior incompleto	Analista de Produção	3 a 4 anos	menos de 2 horas	2	4
25	masc.	superior completo	Analista de Sistemas	2 a 3 anos	5 a 10 horas	3	5
25	masc.	superior completo	Eng. de SW	mais de 4 anos	5 a 10 horas	3	3
26	masc.	superior incompleto	Desenvolvedor de SW	mais de 4 anos	2 a 5 horas	3	5
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	5
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	5
28	masc.	superior incompleto	Desenvolvedor de SW	3 a 4 anos	2 a 5 horas	2	4
29	fem.	superior incompleto	Consultora de Vendas	mais de 4 anos	menos de 2 horas	3	5
30	fem.	superior completo	Eng. de processo	3 a 4 anos	2 a 5 horas	2	4
33	fem.	superior completo	Tec. Judiciária	3 a 4 anos	menos de 2 horas	3	5
36	masc.	superior incompleto	Assis. Administrativo	2 a 3 anos	menos de 2 horas	5	5
36	masc.	superior completo	Representante comercial	3 a 4 anos	menos de 2 horas	2	4

Tabela 7.6: Dados estatísticos obtidos para o quesito Memorização

<b>Memorização</b>			
<b>Método de interação</b>	<b>número de amostras</b>	<b>média</b>	<b>desvio-padrão</b>
<b>Mandos</b>	20	4,1	0,9
<b>Normal</b>	20	3,3	0,9
<b>Mandos – Normal</b>	20	0,8	1,4
<b>Diferença entre pares</b>	<b>n</b>	<b>soma rank</b>	<b>média rank</b>
<b>Positiva</b>	11	102,5	9,32
<b>Negativa</b>	4	17,5	4,38
<b>Zero</b>	5		
<b>valor-p (Wilcoxon signed-rank test)</b>		<b>0,0075</b>	

O valor- $p$  obtido através do teste de comparação de média Wilcoxon signed-rank foi de 0,0075, o que significa que há uma evidência estatística muito elevada de

que a média do quesito memorização obtida pelos aplicativos que faziam uso do *método de interação Mandos* é maior do que as dos aplicativos que não utilizaram o citado método.

A Figura 7.3 ilustra, através do diagrama de caixa (95% CI Notched Skeletal Boxplot) e de diamante (95% CI Mean Diamond), os dados obtidos de maneira estatística.

Através das análises estatísticas dos dados obtidos pelo experimento descrito no capítulo 6, pode-se assumir que a Hipótese III, descrita na seção 4.4 deste trabalho, de que a avaliação dos aplicativos que utilizam o *método de interação Mandos*, no que se diz respeito à memorização, será melhor do que a dos aplicativos que não utilizam o citado método de interação, é verdadeira, o que acarreta em uma melhoria parcial na usabilidade, confirmando parcialmente que o método *de interação Mandos* apresenta uma melhoria de usabilidade nos aplicativos que o utilizam.

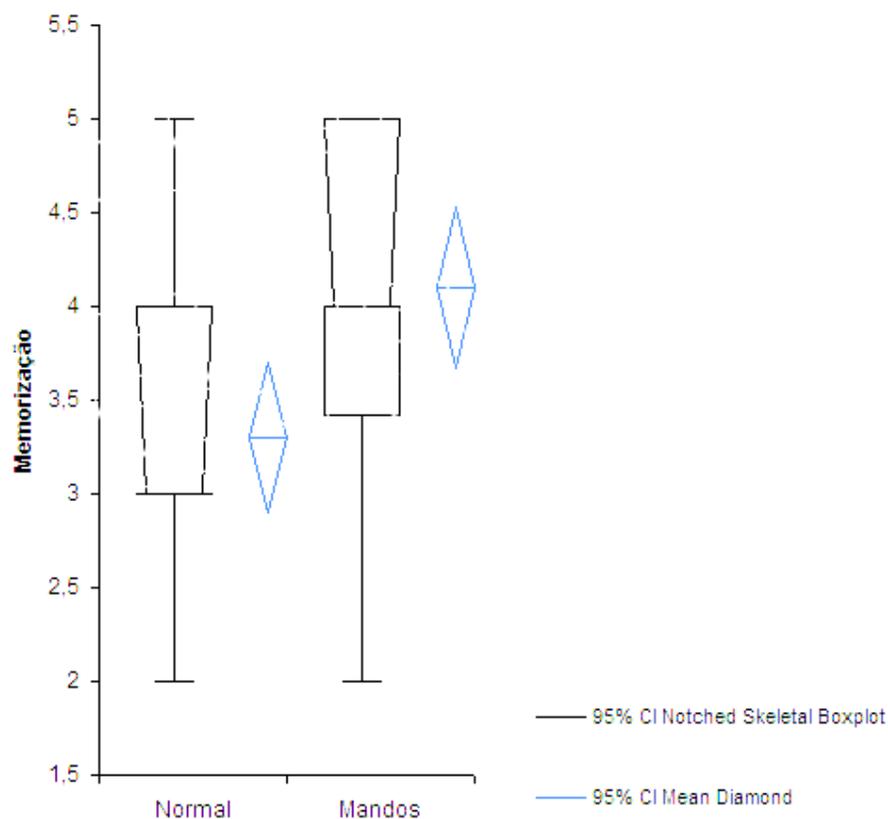


Figura 7.3: Diagrama de Caixa e de Diamante dos valores do quesito Memorização

## 7.4 Resultados do quesito Redução de Erros

A Tabela 7.7 contém uma coletânea de informações sobre os participantes da pesquisa associado com as notas dadas para o quesito redução de erros para os aplicativos utilizados que faziam o uso do método de interação Mandos (coluna “com o uso”) e os que não faziam uso (coluna “sem o uso”).

Os dados resultantes da análise estatística, relativos ao requisito redução de erros, estão apresentados na Tabela 7.8.

Tabela 7.7: Notas atribuídas ao quesito Redução de Erros para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos

Dados dos Participantes						redução de erros	
idade	sexo	escolaridade	profissão	tempo de uso de app.	horas de uso por semana	sem o uso	com o uso
16	fem.	ens. médio incompleto	Estudante (eletrotécnica)	3 a 4 anos	2 a 5 horas	3	4
18	fem.	ens. médio completo	Aux. Administrativo	2 a 3 anos	menos de 2 horas	4	3
19	fem.	superior incompleto	Estudante (direito)	3 a 4 anos	2 a 5 horas	3	4
20	masc.	superior incompleto	Desenvolvedor de SW	1 a 2 anos	menos de 2 horas	3	4
21	fem.	superior incompleto	Estudante (medicina)	mais de 4 anos	menos de 2 horas	3	4
21	fem.	superior incompleto	Estudante (letras)	mais de 4 anos	menos de 2 horas	4	3
23	fem.	superior incompleto	Servidora pública	mais de 4 anos	menos de 2 horas	3	2
24	fem.	superior completo	Eng. de Qualidade	mais de 4 anos	menos de 2 horas	4	5
24	masc.	superior incompleto	Analista de Produção	3 a 4 anos	menos de 2 horas	4	4
25	masc.	superior completo	Analista de Sistemas	2 a 3 anos	5 a 10 horas	2	3
25	masc.	superior completo	Eng. de SW	mais de 4 anos	5 a 10 horas	3	2
26	masc.	superior incompleto	Desenvolvedor de SW	mais de 4 anos	2 a 5 horas	2	3
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	4
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	3
28	masc.	superior incompleto	Desenvolvedor de SW	3 a 4 anos	2 a 5 horas	3	3
29	fem.	superior completo	Consultora de Vendas	mais de 4 anos	menos de 2 horas	2	4
30	fem.	superior completo	Eng. de processo	3 a 4 anos	2 a 5 horas	2	4
33	fem.	superior completo	Tec. Judiciária	3 a 4 anos	menos de 2 horas	4	4
36	masc.	superior incompleto	Assis. Administrativo	2 a 3 anos	menos de 2 horas	4	4
36	masc.	superior incompleto	Representante comercial	3 a 4 anos	menos de 2 horas	3	3

Tabela 7.8: Dados estatísticos obtidos para o quesito Redução de Erros

<b>Redução de Erros</b>			
<b>Método de interação</b>	<b>número de amostras</b>	<b>média</b>	<b>desvio-padrão</b>
<b>Mandos</b>	20	3,5	0,8
<b>Normal</b>	20	3,1	0,7
<b>Mandos – Normal</b>	20	0,4	0,9
<b>Diferença entre pares</b>	<b>n</b>	<b>soma rank</b>	<b>média rank</b>
<b>Positiva</b>	10	79,0	7,90
<b>Negativa</b>	4	26,0	6,50
<b>Zero</b>	6		
<b>valor-p (Wilcoxon signed-rank test)</b>		<b>0,0520</b>	

O valor- $p$  obtido através do teste de comparação de média Wilcoxon signed-rank foi de 0,0520, o que significa que há evidência insuficiente contra a hipótese nula e ela não deve ser rejeitada, ou seja, não há evidências suficientes de que a média do quesito redução de erros obtida pelos aplicativos que faziam uso do *método de interação Mandos* é maior do que as dos aplicativos que não utilizaram o citado método.

A Figura 7.4 ilustra, através do diagrama de caixa (95% CI Notched Skeletal Boxplot) e de diamante (95% CI Mean Diamond), os dados obtidos de maneira estatística.

Através das análises estatísticas dos dados obtidos pelo experimento descrito no capítulo 6, pode-se assumir que a Hipótese IV, descrita na seção 4.4 deste trabalho, de que a avaliação dos aplicativos que utilizam o *método de interação Mandos*, no que se diz respeito à redução de erros, será melhor do que a dos aplicativos que não o citado método é falsa, este requisito se torna nulo na análise geral do aprimoramento da usabilidade pelo *método de interação Mandos*.

Acredita-se que a avaliação negativa do quesito redução de erros seja causada pelo impacto do primeiro contato com *método de interação Mandos* o que causou a princípio alguns erros na realização das tarefas. Este quesito necessita de uma avaliação a longo prazo.

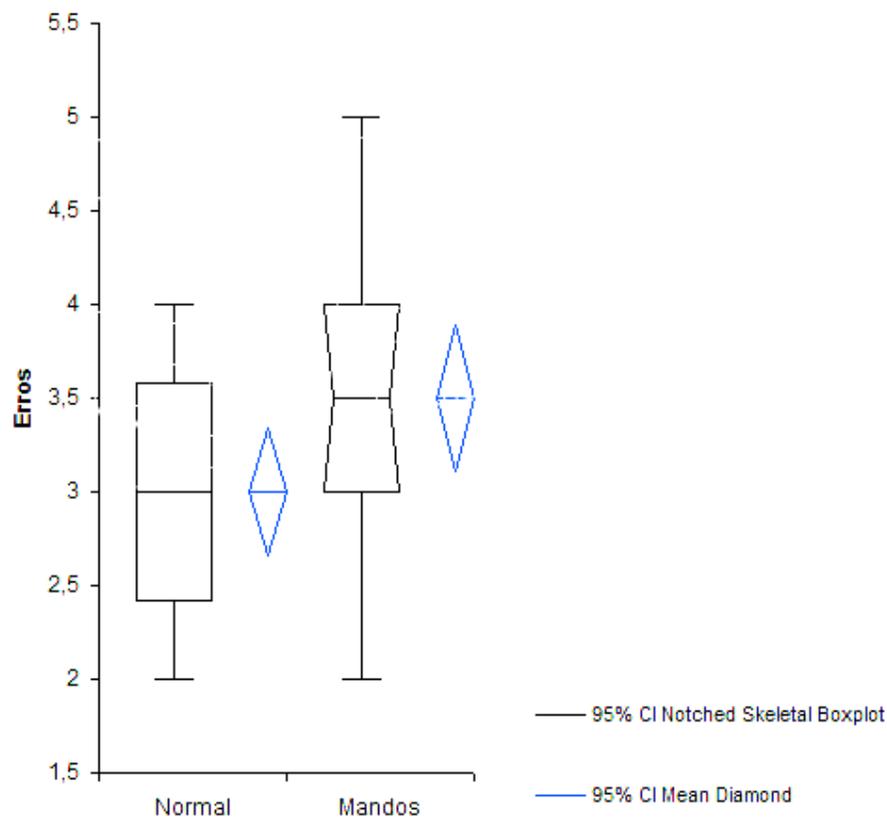


Figura 7.4: Diagrama de Caixa e de Diamante dos valores do quesito Redução de Erros

## 7.5 Resultados do quesito Satisfação

A Tabela 7.9 contém uma coletânea de informações sobre os participantes da pesquisa associado com as notas dadas para o quesito satisfação para os aplicativos utilizados que faziam o uso do método de interação Mandos (coluna “com o uso”) e os que não faziam uso (coluna “sem o uso”).

Tabela 7.9: Notas atribuídas ao quesito Satisfação para os aplicativos que utilizam e os que não utilizam o Método de interação Mandos

Dados dos Participantes						satisfação	
idade	sexo	Escolaridade	profissão	tempo de uso de app.	horas de uso por semana	sem o uso	com o uso
16	fem.	ens. médio incompleto	Estudante (eletrotécnica)	3 a 4 anos	2 a 5 horas	2	4
18	fem.	ens. médio completo	Aux. Administrativo	2 a 3 anos	menos de 2 horas	2	4
19	fem.	superior incompleto	Estudante (direito)	3 a 4 anos	2 a 5 horas	2	4
20	masc.	superior incompleto	Desenvolvedor de SW	1 a 2 anos	menos de 2 horas	4	4
21	fem.	superior incompleto	Estudante (medicina)	mais de 4 anos	menos de 2 horas	2	4
21	fem.	superior incompleto	Estudante (letras)	mais de 4 anos	menos de 2 horas	4	2
23	fem.	superior incompleto	Servidora pública	mais de 4 anos	menos de 2 horas	2	4
24	fem.	superior completo	Eng. de Qualidade	mais de 4 anos	menos de 2 horas	2	4
24	masc.	superior incompleto	Analista de Produção	3 a 4 anos	menos de 2 horas	3	4
25	masc.	superior completo	Analista de Sistemas	2 a 3 anos	5 a 10 horas	3	5
25	masc.	superior completo	Eng. de SW	mais de 4 anos	5 a 10 horas	3	5
26	masc.	superior incompleto	Desenvolvedor de SW	mais de 4 anos	2 a 5 horas	3	4
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	3	5
27	masc.	superior completo	Eng. Eletricista	2 a 3 anos	menos de 2 horas	2	3
28	masc.	superior incompleto	Desenvolvedor de SW	3 a 4 anos	2 a 5 horas	3	4
29	fem.	superior completo	Consultora de Vendas	mais de 4 anos	menos de 2 horas	4	5
30	fem.	superior completo	Eng. de processo	3 a 4 anos	2 a 5 horas	2	4
33	fem.	superior completo	Tec. Judiciária	3 a 4 anos	menos de 2 horas	3	5
36	masc.	superior incompleto	Assis. Administrativo	2 a 3 anos	menos de 2 horas	5	5
36	masc.	superior incompleto	Representante comercial	3 a 4 anos	menos de 2 horas	2	5

Os dados resultantes da análise estatística, relativos ao requisito memorização, estão apresentados na Tabela 7.10.

O valor- $p$  obtido através do teste de comparação de média Wilcoxon signed-rank foi de 0,0003, o que significa que há uma evidência estatística muito elevada de que a média do quesito satisfação obtida pelos aplicativos que faziam uso do *método de interação Mandos* é maior do que as dos aplicativos que não utilizaram o citado método.

A Figura 7.5 ilustra, através do diagrama de caixa (95% CI Notched Skeletal Boxplot) e de diamante (95% CI Mean Diamond), os dados obtidos de maneira estatística.

Através das análises estatísticas dos dados obtidos pelo experimento descrito no capítulo 6, pode-se assumir que a Hipótese V, descrita na seção 4.4 deste

trabalho, de que a avaliação dos aplicativos que utilizam o *método de interação Mandos*, no que se diz respeito à satisfação, será melhor do que a dos aplicativos que não utilizam o citado método de interação, é verdadeira, o que acarreta em uma melhoria parcial na usabilidade, confirmando parcialmente que o método *de interação Mandos* apresenta uma melhoria de usabilidade nos aplicativos que o utilizam.

Tabela 7.10: Dados estatísticos obtidos para o quesito Satisfação

<b>Satisfação</b>			
<b>Método de interação</b>	<b>número de amostras</b>	<b>média</b>	<b>desvio-padrão</b>
<b>Mandos</b>	20	4,2	0,8
<b>Normal</b>	20	2,8	0,9
<b>Mandos – Normal</b>	20	1,4	1,1
<b>Diferença entre pares</b>	<b>n</b>	<b>soma rank</b>	<b>média rank</b>
<b>Positiva</b>	17	159,5	9,38
<b>Negativa</b>	1	11,5	11,5
<b>Zero</b>	2		
<b>valor-p (Wilcoxon signed-rank test)</b>		<b>0,0003</b>	

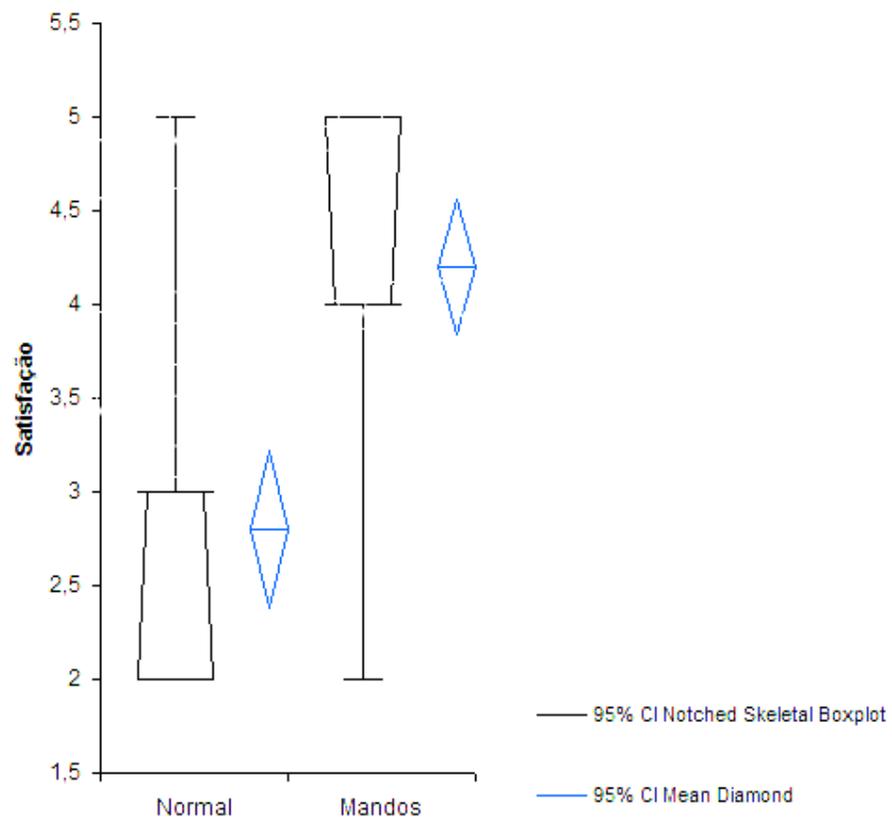


Figura 7.5: Diagrama de Caixa e de Diamante dos valores do quesito Satisfação

## 7.6 Resultados gerais do Método de interação Mandos

Para se comprovar a hipótese que o uso do *método de interação Mandos* resultaria em uma melhoria na usabilidade de um aplicativo embarcado em um telefone móvel, adotou-se a estratégia de dividir a usabilidade em quesitos e então mensurar se houve alguma melhoria em cada um dos quesitos e com estas informações poder inferir a melhoria da usabilidade em geral.

Dos cinco quesitos analisados, que são: intuitividade, eficiência, memorização, redução de erros e satisfação, os aplicativos que fizeram uso do *método de interação Mandos* em quatro quesitos obtiveram média superior com evidência estatística elevada e um dos itens não apresentou evidência estatística de aprimoramento e nem de desaprovação.

Para comprovação da Hipótese principal, descrita na seção 4.4 deste trabalho, que diz que o uso do *método de interação Mandos* em aplicativos embarcados em telefones móveis resultará em um aprimoramento da usabilidade, estabeleceu-se que em pelo menos 4 quesitos deveria alcançar uma avaliação superior, o que foi atingido e com isto pode-se declarar como verdadeira a Hipótese principal.

Como intuito de se aprofundar ainda mais nas impressões resultantes do uso do *método de interação Mandos*, através das entrevistas realizadas com os participantes ao final do experimento pôde-se colher alguns dados importantes que estão dispostos a seguir.

Sobre o uso do *método de interação Mandos*:

- 90% dos entrevistados demonstraram interesse em utilizar o método de interação Mandos no seu telefone móvel;
- 5% dos entrevistados demonstraram interesse em utilizar com a ressalva de que todo o telefone móvel deveria seguir o método de interação Mandos;
- 5% dos entrevistados não gostariam de ter o método de interação Mandos em qualquer aplicativo instalado no seu telefone móvel;
- 40% dos entrevistados afirmam que se sentiram um pouco confusos com o uso do método de interação Mandos.

Um dos objetivos do uso do método de interação Mandos é apresentar uma forma dos aplicativos adaptarem-se ao usuário, uma forma alternativa de uso desta idéia é de haver um aplicativo que pudesse definir os valores da matriz de transição de tarefas de forma simples e viável ao usuário do telefone móvel, podendo o mesmo realizar a citada operação em um computador de mesa e depois carregar para o telefone celular, com base nesta idéia apresenta ao entrevistado, constatou-se que:

- 85% dos entrevistados fariam uso de um aplicativo desenvolvido para customizar, personalizar os aplicativos a serem instalados no telefone móvel;

Durante a entrevista vários comentários foram anotados, dentre os principais e mais comuns destaca-se que a grande maioria dava exemplos e situações de como o *método de interação Mandos* poderia auxiliar no seu uso cotidiano do seu telefone móvel, não apenas em aplicativo embarcado, cujo está o foco deste trabalho, mais na construção geral da interface do usuário do telefone móvel,

Um comentário interessante do participante foi que “toda inovação é bem-vinda, desde que não gere um custo a mais na aquisição do celular”, não foi feito um estudo sobre quanto o uso do método de interação Mandos poderia aumentar o preço do telefone móvel, mas como o método se restringe apenas ao software e o

desenvolvimento do *framework* proposto foi justamente para facilitar a adaptação de um aplicativo já desenvolvido, acredita-se que o impacto seja mínimo, principalmente se a concepção do mesmo se basear desde o início neste método de interação.

Um dos questionamentos relevantes foi se após o aplicativo embarcado em um telefone móvel adaptar-se a um usuário e o citado telefone mudar de dono, talvez o novo dono ache a disposição das tarefas péssimas. Sobre esta situação o que pode se dizer é que o aplicativo continuará a se adaptar ao novo dono e que pode ser planejado uma forma dos valores da matriz de transição de tarefas serem atualizados com os valores padrões iniciais, mas este trabalho não contemplou o estudo desta situação para propor uma solução adequada.

## 7.7 Limitações do Experimento

Os resultados apresentados podem ter alguma distorção da realidade devido às limitações do experimento, dentro as quais o tempo de uso do aplicativo durante o experimento, o ideal seria que o participante do experimento pudesse fazer uso dos aplicativos avaliados no seu cotidiano, uma avaliação em longo prazo poderia acarretar em resultados diferentes, o tipo do aplicativo utilizado no experimento também foi uma limitação do mesmo, o resultado da avaliação do experimento pode ser alterado de acordo com o tipo de aplicativo a ser avaliado, citam-se ainda o efeito de expectativa do sujeito e efeito de expectativa do experimentador, que são detalhados nos sub-itens seguintes.

### 7.7.1 Efeito de expectativa do sujeito – efeitos placebo e Hawthorne

Segundo Wainer (2007), o efeito placebo é muito conhecido na medicina, e diz que se você de um remédio inócuo para um paciente, ele vai dizer que seus sintomas melhoraram. A expectativa que o paciente causa a melhora. Um efeito similar ao placebo e talvez mais relevante para o corrente experimento é o efeito Hawthorne, que diz que pode haver um efeito positivo apenas pelo fato dos sujeitos saberem que estão sendo estudados/observados.

O caso que gerou a teoria do efeito foi numa fábrica chamada Hawthorne, onde se estudou o efeito de níveis de iluminação na produtividade dos trabalhadores, e descobriu-se que a produtividade dos trabalhadores aumentava independentemente de mudanças no nível de luminosidade – a teoria é que sabendo que estavam sendo observados melhorou a produtividade dos trabalhadores. No corrente experimento os usuários entrevistados podem ter melhorado seu desempenho pelo efeito Hawthorne.

### 7.7.2 Efeito de expectativa do experimentador

Segundo Wainer (2007), o efeito de expectativa do experimentador acontece em alguns exemplos onde o pesquisador interage intensamente como sujeito, e as crenças do experimentador causam um efeito no sujeito (ou pelo menos nos testes realizados pelo sujeito). Um exemplo claro desse efeito acontece quando o pós-teste requer alguma avaliação subjetiva do pesquisador, como foi o caso do experimento descrito neste trabalho, se o pesquisador sabe de quem são os testes submetidos à

intervenção, e ele espera que a intervenção seja positiva, então ele pode “melhorar” as notas do pós-teste, mesmo inconscientemente.

Um outro exemplo do fenômeno, mesmo quando não há o componente subjetivo no pós-teste, é o efeito Pigmaleão ou Rosenthal em educação – quando professores foram (falsamente) informados que suas classes tinham alunos mais inteligentes que a média, os alunos tiveram resultados muito melhores que alunos similares, mas cujos professores não receberam a falsa informação. A teoria propõe que o experimentador passa sinais inconscientes que acabam influenciando os sujeitos. O efeito de expectativa do experimentador também pode ser muito relevante para experimentos em ciência da computação. Se o experimentador é o criador de um sistema, ele pode passar aos sujeitos sinais que indicam sua expectativa que seu sistema é útil e bom, e, é claro, se a avaliação do pós-teste tiver algum aspecto subjetivo.

# Capítulo 8

## Conclusão

A usabilidade na telefonia móvel é hoje alvo de diversos estudos e desenvolvimentos de técnicas para aprimoramento devido ao grande potencial econômico e comercial e também por ser uma tecnologia e área incipiente.

Este trabalho procurou dar a sua contribuição para esta área da computação, propondo uma nova forma de interação com o usuário em telefones móveis, focada em aprimorar a usabilidade dos aplicativos embarcados. Cita-se também como contribuição deste trabalho a criação e desenvolvimento do *framework Mandos*, que possibilita a criação de aplicativos embarcados que empreguem o *método de interação Mandos*.

A hipótese principal elaborada a ser comprovado pelo experimento conduzido neste trabalho é que a avaliação da usabilidade dos aplicativos que utilizassem o *método de interação Mandos* seria melhor do que as dos aplicativos que não utilizassem o citado método de interação.

Após a realização do experimento que mensurou a usabilidade de aplicativos desenvolvidos com e sem o *método de interação Mandos*, levando em conta as suas limitações, pôde-se inferir que o método de interação proposto apresentou indícios de um aprimoramento na usabilidade dos aplicativos embarcados na telefonia móvel, obtendo avaliações com evidências estatísticas elevadas de superioridade em quatro dos cinco quesitos avaliados que compõe a usabilidade: intuitividade, eficiência, memorização e satisfação.

Assim sendo, a principal conclusão deste trabalho é que o uso do *método de interação Mandos* em aplicativos embarcados na telefonia móvel apresenta um sinal de evidência no aprimoramento na usabilidade, sendo necessário um experimento mais profundo e detalhado, para admitir que o uso do *método de interação Mandos* resulta em uma melhoria da usabilidade de um aplicativo embarcado em um telefone móvel.

A usabilidade em dispositivos de telefonia móvel é um tópico relativamente novo de pesquisa. Conseqüentemente, possui muitas oportunidades para melhoramentos adicionais. Os trabalhos seguintes são considerados como extensões futuras:

- O desenvolvimento e avaliação de outros tipos de aplicativos embarcados em telefones móveis utilizando o método de interação Mandos, como

aplicações utilizadas diariamente e com muita frequência, como, por exemplo, aplicações de negócios e empresariais.

- Eliminar a limitação do uso do *método de interação Mandos*, que no atual trabalho está restrito dentro do escopo de aplicações embarcadas, desenvolvidas com a tecnologia Java e fazer com que o método de interação seja utilizado na criação de toda e qualquer interface de um telefone móvel e em outras linguagens, como Symbian, Brew, etc...
- A elaboração de um produto em que o usuário pudesse determinar os valores desejados da matriz de probabilidades de transição entre tarefas, este produto poderia ser um aplicativo a ser executado em um computador de mesa, onde o usuário seria capaz de graficamente escolher as ordens das principais tarefas.
- O experimento conduzido apresenta algumas limitações, uma das quais é a duração do tempo do experimento. Para a correta avaliação da adaptação de um aplicativo a um usuário específico é necessário um experimento em longo prazo. Uma extensão futura seria a realização de experimentos dando a oportunidade de que os participantes tenham mais tempo para a avaliação.



## Referências Bibliográficas

ANALYSE-IT. **Statistical analysis and statistics software for Microsoft Excel**. Acessível em: <<http://www.analyse-it.com/>>. Acessado em: 01 de setembro de 2007.

ANATEL. **PASTE/2000-Perspectivas para Ampliação e Modernização do Setor de Telecomunicações para o período 2000/2005**. ANATEL-Agência Nacional de Telecomunicações. Brasília, 2000.

ARA, A. B., MUSETTI, A.V., SCHNEIRMAN, B. **Introdução à Estatística**. São Paulo: Instituto Mauá de Tecnologia, Edgard Blucher Ltda, 2003.

BAX, M. Peixotto. **Introdução às Linguagens de Marcas "Introduction to Markup Languages"**. Universidade Federal de Minas Gerais. Brasil, abril, 2000.

BETIOL, Adriana H. **Avaliação de Usabilidade para os Computadores de Mão: Um Estudo Comparativo entre três Abordagens para Ensaios de Interação**. Tese de Doutorado em Engenharia de Produção. UFSC, 2004.

BREWSTER, S. **Overcoming the lack of screen space on mobile computers**. Personal and Ubiquitous Computing 6 (3) May, 2002.

CARROL, J.W. & THOMAS, J.C. **Fun**. SIGCHI Bulletin. January 1988, Vol.19 No.3, 1988.

CAULTON, D. A. **Relaxing the homogeneity assumption in usability testing**. Behaviour & Information Technology, vol. 20, nº 1, 1-7, 2001.

CHINCHOLLE, D., GOLDSTEIN, M., NYBERG, M., ERIKSSON, M. **Lost or found? A usability evaluation of a mobile navigation and location-based service**. Proceedings of Mobile HCI 2002 - Fourth International Symposium on Human Computer Interaction with Mobile Devices. Pisa, Italy, September 2002.

COOPER, 1995. **Performance Measurement Handbook**. National Physical Laboratory, Teddington, Middlesex, U.K. 1995.

DUMAS, J.S., REDISH, J.C. **A Practical Guide to Usability Testing**. England: Intellect, 1999.

EISENSTEIN, Jacob, VANDERDONCKT, Jean and PUERTA, Angel. **Applying Model-Based Techniques to the Development of UIs for Mobile Computers**. ACM Press, 2001.

ERICSSON, T. Chincholle, D., Goldstein, M. **Both the celular phone and the service impact WAP usability**. Proceedings of IHM-HCI 2001, Volume II, HCI in Practice, by J. Vanderdonckt, A. Blanford and A. Derycke (eds). Lille, France, 10-14 September, 2001.

FERREIRA, K. Gomes. **Testes de Usabilidade**. 2004. 60 f. Monografia (3) - Curso de Especialização em Informática: Ênfase em Engenharia de Software, Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2002.

FONSECA, Enrico. **Ciclo de vida do MIDlet**. Acessível em: <<http://www.imasters.com.br/artigo/3416>>. Acessado em: 22 de agosto de 2007.

GREENE, S. & FINNEGAN, J. **Usability of Mobile Devices and Intelligently Adapting to a User's Needs**. Waterford Institute of Technology, 2003.

HAITANI, R and BERGMAN, E. Designing the PalmPilot: **A Conversation with Rob Haitani, Chapter 4 in Information Appliances and Beyond**, Eric Bergman (Ed.), Morgan Kaufmann, 2000.

HILBERT, David and REDMILES, David. **Extracting Usability Information from User Interface Events**. ACM Press, 2000.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. **ISO 9241 Ergonomic Requirements for Office Work with Visual Display Terminals (VDTs) – Part 11: Usability Principles**. International Standard, Suécia, 1998.

JCP. JSR 238: Mobile Internationalization API. Acessível em: <<http://jcp.org/en/jsr/detail?id=238>>. Acessado em: 23 de agosto de 2006.

KIM, H.; KIM, J.; LEE, Y.; CHAE, M.; CHOI, Y. **An Empirical Study of the Use Contexts and Usability Problems in Mobile Internet**. Proceedings of the 35th Annual Hawaii International Conference on Systems Sciences, 2002.

KIM, J. H. **Culturally Adapted Mobile Phone Interface Design: Correlation between Categorization Style and Menu Structure**. HCI Mobile' 05, ACM Press, 2005.

KLOCKAR, T.; CARR, D. A.; HEDMAN, A.; JOHANSSON, T.; BENGTSSON, F. **Usability of Mobile Phones**. Luled University of Technology, 2003.

LARMAN, C. **Utilizando UML e padrões**. Porto Alegre: Bookman, 2000.

MACKENZIE, I.S., SOUKOREFF, R.W. **Text entry for mobile computing: Models and methods, theory and practice**. Human-Computer Interaction, 17, 147-198. 2002.

MACLEOD, M., BOWDEN, R., BEVAN, N., CURSON, I. **The MUSiC performance measurement method**. Behaviour & Information Technology, vol. 16, nº 4/5, 279-293. 1997.

MARTINS, Renato. **Tutorial de programação J2ME, Parte 2 - Plataforma Java 2 Micro Edição (J2ME)**. Acessível em: <[http://www.wirelessbrasil.org/wirelessbr/colaboradores/corbera\\_martins/j2me\\_02.html](http://www.wirelessbrasil.org/wirelessbr/colaboradores/corbera_martins/j2me_02.html)> Acessado em: 15 de maio de 2006.

MAYBURY, M. T. **User interfaces for all: concepts, methods and tools**. Lawrence Erlbaum Associates, Inc., USA, 2001.

MEYER, B. **Object-oriented software construction**. Englewood Cliffs: Prentice Hall, 1988.

MOHAGEG, M. F., WAGNER, A. **Design consideration for Information Appliances**. In: Bergman, Eric. Information Appliances and Beyond: interaction design for consumer products. San Diego, CA: Academic Press, 2000.

MONTGOMY, D. C, RUNGER, C. R., HUBELE, N. F. **Estatística Aplicada à Engenharia**. Editora LTC. Segunda Edição. Rio de Janeiro, 2004.

MUCHOW, Jonh W. **Core J2ME Technology & MIDP**. Sun Microsystems Press, USA., 2002.

NEMER, Amarílis C. **Estudo de Usabilidade em Telefones Celulares**. Trabalho Final de Mestrado Profissional em Computação. UNICAMP, 2006.

NIELSEN, J. LANDAUER, T.K. **A mathematical model of the finding of usability problems**. INTERCHI'93. Amsterdam, the Netherlands, 24-29 April, 1993.

NIELSEN, J. **Usability Engineering**. San Francisco, CA: Morgan Kaufmann, 1993  
PALEN, Leysia. **Beyond the handset: designing for wireless communications usability**, ACM Press, 2002.

NIELSEN, J. **230 Tips and Tricks for a Better Usability Test**. Nielsen Norman Group, 2003.

NIELSEN, J. **Return on Investment for Usability**. Jakob's Nielsen Alertbox. Acessível em: <<http://www.useit.com/alertbox/20030107.html>>. Acessado em: 15 de maio de 2006.

PESSOA, C. A. C., RAMALHO, G. L. e BATTAIOLA, A. L.: **wGEM: um Framework de Desenvolvimento de Jogos para Dispositivos Móveis**. UFPE / UFSCar, 2003.

QIU, M. K.; ZHANG, K.; HUANG, M. **An Empirical Study of Web Interface Design on Small Display Devices**. 2004.

RAO, D. O. **A study of input devices used on personal digital assistants (PDAs)**. Serco Usability Services, London, 2000.

REINALDO, Francisco Antonio Fernandes. **Definição e aplicação de um framework para desenvolvimento de redes neurais modulares e heterogêneas**. 86 f. Dissertação (Mestrado) - Universidade Federal de Santa Catarina. Florianópolis, 2003.

RIBEIRO JUNIOR, Aloísio. **Um Estudo para a Introdução de Testes de Usabilidade no Desenvolvimento de Sistemas de Engenharia**. UFMG, 2003.

RISCHPATER, R. **Wireless Web Development**. Berkeley, CA: Apress, 2000.

ROCHA, H. V., BARANAUSKAS, M.C. **Design e Avaliação de Interfaces Humano Computador**. Campinas: NIED, 2003.

RODRIGUES, E. B. **Métodos de Testes de Usabilidade em Aplicações em Celular**. Monografia de especialização em Sistemas Móveis e Convergentes em Telefonia Celular. UEA – Universidade do Estado do Amazonas, 2007.

RUBIN, Jeffrey. **Handbook of Usability Testing: How to Plan, Design and Conduct Effective Tests**. New York: John Wiley & Sons, Inc., 1994.

SACHER, H. and LOUDON, G. (2002) **Uncovering the New Wireless Interaction Paradigm**. Interactions ACM. January and February 2002.

SHACKEL, B. **Human Factors for Informatics Usability**. University Press, Cambridge, 1991.

SHEN, J.; SHEN, X. **User Requirements in Mobile Systems**. Seventh Americas Conference on Information Systems, 2001.

SHIPLE, John. **Information Architecture Tutorial**. Acessível em: <[http://www.webmonkey.com/webmonkey/design/site\\_building/tutorials/tutorial1.html](http://www.webmonkey.com/webmonkey/design/site_building/tutorials/tutorial1.html)>. Acessado em: 21 de junho de 2006.

SILVA, R. P. **Suporte ao desenvolvimento e uso de frameworks e componentes**. Porto Alegre: PPGC da UFRGS, 2000.

SOMEREN, Maarten. Barnard, Yvonne. Sandberg, Jacobijn. **The think aloud method**. London. Academic Press. Department of Social Science Informatics. 1994.

SONY ERICSSON. **Introducing XML Parsing with JSR 172**. Acessível em: <[http://developer.sonyericsson.com/site/global/techsupport/tipstrickscode/java/p\\_xml\\_parsing\\_jsr172.jsp](http://developer.sonyericsson.com/site/global/techsupport/tipstrickscode/java/p_xml_parsing_jsr172.jsp)>. Acessado em 22 de maio de 2006.

SPOOL, J., SCHROEDER, W. **Testing Web sites: five users is nowhere near enough**. CHI 2001, Extended Abstracts. Seattle, March 31- April 5, 2001.

SPYRIDAKIS, J.H. **Conducting Research in Technical Communication: The Application of True Experimental Designs**. Technical Communication, 39, 4, 607-624, 1992.

SUN MICROSYSTEMS, INC. **Sun Java Wireless Toolkit**. Acessível em: <<http://java.sun.com/j2me>>. Acessado em 30 de maio de 2006.

THIMBLEY, Harold. **Usability analysis with Markov models**. ACM Press, 2001;

TOLKIEN, J. R. R. **O Silmarilion**. Editora Martins Fontes, 4. edição, 2003.

UFRGS. **Connected, Limited Device Configuration e Mobile Information Device Profile**. Acessível em: <<http://www.inf.ufrgs.br/propar/disc/inf01008/trabalhos/sem01-1/t2/pitoni/#top>>. Acessado em: 21 de junho de 2006.

VIRZI, R. A. **Refining the test phase of usability evaluation: how many subjects is enough?** Human Factors 34 (4), 1992.

WAINER, J. **Pesquisa quantitativa e qualitativa em ciência da computação**. Atualização em Informática 2007 Sociedade Brasileira de Computação, Rio de Janeiro, 2007.

WEISS, S., KEVIL, D., MARTIN, R. **Wireless phone usability findings**. Usable Products, New York, January, 2001.

WINTER, E.M.W. **Introdução à Bioestatística**. Departamento de Estatística, Universidade Federal do Paraná. 2002.

WOOLRYCH, A., COCKTON, G. **Why and when five test users aren't enough**. Proceedings of IHM-HCI 2001 Conference. Toulouse, France, 2001.

# Anexo A

## Plano de Teste

### A.1 Propósito do Teste

O propósito deste teste é verificar a adaptação, desempenho alcançado pelo usuário, assim como o entendimento do novo modo operacional de interação com o usuário proposta pelo uso do framework Mandos no desenvolvimento de um aplicativo embarcado no celular. Será medido o grau de satisfação do usuário em relações aos itens que formam a usabilidade segundo Nielsen, assim como serão identificadas as dificuldades envolvendo a utilização do aplicativo, assim como a sua disposição em utilizar mais a fundo o método de interação Mandos.

### A.2 Declaração dos Problemas

1. A usabilidade do aplicativo melhorou com o uso do framework Mandos?
2. O usuário prefere usar o aplicativo que utiliza o método de interação Mandos?

### A.3 Perfil do Usuário

Serão utilizados vinte participantes. Os participantes devem ter de 15 a 40 anos de idade, nível médio (completo ou não) ou superior (completo ou não) e mais de um ano de uso de telefone celular.

### A.4 Metodologia

O teste será realizado com a finalidade de garantir a usabilidade do produto e será composto das seguintes partes:

1. Cada participante será devidamente cumprimentado pelo avaliador, será orientado a se sentar e tentar se sentir confortável e relaxado. O participante será orientado a preencher um pequeno questionário para identificação de seu perfil (Questionário para Identificação do Perfil do Participante).

2. O participante receberá um script introdutório de orientação do teste (Script de Orientação), explicando o propósito e objetivos do teste. Deve ser reforçado que o produto é o centro da avaliação e não o participante e que as tarefas devem ser executadas de forma bastante confortável.

3. Depois de passadas as orientações, será permitido que o participante utilize os aplicativos serem testado livremente por cinco minutos. Logo depois lhe será entregue a lista de tarefas (Lista de Tarefas). O avaliador irá requisitar que o participante verbalize suas dúvidas, pois isto ajudará ao avaliador anotar a ocorrência e a razão de problemas. Durante o teste, os acontecimentos observados pelo avaliador serão registrados em formulário próprio (Coleta de Dados pelo Avaliador).

4. Depois de completadas todas as tarefas, o participante preencherá um questionário de avaliação do sistema cuja finalidade é coletar informações preferenciais do participante (Questionário de Avaliação do Sistema pelo Participante).

5. Depois, o participante será questionado pelo avaliador em uma sessão de questionamento do participante. Serão discutidas percepções subjetivas de usabilidade do participante acerca do sistema, realizados comentários globais sobre o desempenho do participante e problemas encontrados. O participante poderá comentar sobre o teste abertamente, permitindo uma coleta de informações complementares (Tópicos para Questionamento).

6. Depois da sessão de questionamento do participante, será agradecida a colaboração do participante.

Observação: o avaliador estará de posse do Roteiro do Avaliador para o orientá-lo na condução do teste.

## A.5 Lista de Tarefas

Segue uma lista de tarefas preliminar para o teste de usabilidade do sistema:

- Iniciar o despertador, verificar o status do despertador e sair;
- Iniciar o despertador, ativar o alarme e sair;
- Iniciar o despertador, desativar o alarme e sair;
- Iniciar o despertador, ativar o alarme, configurar hora (8:00 AM) e sair;
- Iniciar o despertador, ativar o alarme, configurar hora (7:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta e sexta) e sair;
- Iniciar o despertador, configurar hora (6:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta, sexta e sábado), definir melodia e sair;
- Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a mais) e sair;
- Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a menos), habilitar pausa automática e sair;
- Iniciar a calculadora, somar 42 com 33, ver o resultado e sair;
- Iniciar a calculadora, multiplicar 56 com 22, ver o resultado, salvar o resultado na memória, somar 47 com 11, somar o resultado obtido com o armazenado na memória e sair;
- Iniciar a calculadora, abrir a memória, limpar a memória, somar 30 com 65 e entrar no modo conversor, voltar para calculadora e sair;
- Iniciar a calculadora, ir para o modo ampliado, tirar a raiz quadrada de 45, salvar na memória, ir para o modo básico, diminuir 32 de 14,5 e somar com o resultado armazenado na memória e sair;

- Iniciar a calculadora, realizar a operação  $20 / 45$ , ver o resultado, limpar a tela e sair;

## **A.6 Ambiente de Teste / Equipamento**

Será registrado, através de algumas fotografias, a realização dos testes pelo usuário. As fotografias serão realizadas durante a execução dos testes para registro das condições do ambiente e usuário.

O ambiente para o teste será um local tranqüilo, livre de perturbações externas que tire a atenção do usuário da realização das tarefas e de maneira a utilizar o telefone celular de maneira eficaz.

O telefone móvel a ser utilizados nos testes será o BenQ-Siemens SL75.

O protótipo disponibilizado estará pronto para realizar todas as funcionalidades requisitadas na lista de tarefas.

## **A.7 Papel do Avaliador**

O avaliador se sentará ao lado do participante durante a realização do teste e registrará a ocorrência de erros e observações através do formulário Coleta de Dados pelo Avaliador.

O avaliador não poderá ajudar o participante na realização das tarefas. Ele somente poderá orientar se surgir uma questão acerca do procedimento de teste.

## **A.8 Medidas de avaliação**

As seguintes medidas de avaliação serão coletadas e calculadas:

1. Número de dúvidas em cada tarefa por participante;
2. Número de erros cometidos na realização de cada tarefa por participante;
3. Dados qualitativos, relativos à usabilidade, sobre a utilização do protótipo dos aplicativos desenvolvidos com o conceito de interação Mandos e o uso dos mesmos aplicativos que foram anteriormente desenvolvidos sem o citado conceito;
4. Dados subjetivos sobre a satisfação do participante;
5. Média dos dados qualitativos relativos à usabilidade;
6. Desvio padrão dos dados qualitativos relativos à usabilidade;
7. Média de erros por tarefa;
8. Desvio padrão da quantidade de erros por tarefa.

# Anexo B

## Roteiro do Avaliador

### B.1 Objetivo

O objetivo deste documento é servir como guia para o avaliador da sessão de teste dos aplicativos a serem comparados. Durante o teste, serão verificadas a satisfação e o desempenho alcançado pelos participantes e o entendimento das funções do sistema utilizando os aplicativos confeccionados para os testes. Serão anotados os erros e dificuldades envolvendo a utilização do protótipo em tarefas rotineiras para verificar a validação do método de interação Mandos.

Este roteiro visa coletar os seguintes dados:

1. Obter dados quantitativos sobre o número de erros cometidos durante a execução das tarefas;
2. Obter dados qualitativos sobre a utilização dos aplicativos utilizando o método de interação Mandos, assim como obter os mesmos dados qualitativos para os mesmos aplicativos sem o uso do método citado para efetuar posterior comparação e validação da eficiência do método de interação proposto;
3. Registrar aspectos sobre a satisfação subjetiva do participante.

### B.2 Ambiente de Teste / Equipamento

Será utilizada uma câmera fotográfica para registrar os testes dos usuários.

O ambiente para o teste será um local tranquilo, livre de perturbações externas que tire a atenção do usuário da realização das tarefas e de maneira a utilizar o telefone celular de maneira eficaz.

O telefone móvel a ser utilizados nos testes será o BenQ-Siemens SL75.

O protótipo disponibilizado estará pronto para realizar todas as funcionalidades requisitadas na lista de tarefas.

### B.3 Papel do Avaliador

O avaliador se sentará em uma cadeira ao lado do participante e utilizará uma prancheta com o formulário para Coleta de Dados pelo Avaliador para anotar os detalhes do teste.

O avaliador não poderá ajudar o participante na realização das tarefas. Ele somente poderá orientar caso surja uma questão acerca do procedimento de teste.

## **B.4 Perfil do Participante**

Serão utilizados trinta participantes. Os participantes devem ter de 15 a 65 anos de idade, nível médio (completo ou não) ou superior (completo ou não) e mais de um dois anos de uso de telefone celular.

## **B.5 Aplicativos a serem testados**

Serão apresentados dois aplicativos a serem testados:

### **B.5.1 Despertador**

Aplicativo que consiste em um despertador programável. Dentre as tarefas a serem executadas pelo aplicativo estão:

### **B.5.2 Calculadora**

Aplicativo que possui uma série de recursos para realização de cálculos.

## **B.6 Protocolos e Procedimentos**

1. O avaliador recebe o participante, o cumprimenta e o convida a se sentar e se sentir confortável e relaxado.
2. O avaliador entrega ao participante o Questionário para Identificação do Perfil do Participante.
3. Após completar o questionário, o participante recebe o Script de Orientação do teste. O avaliador lê o script junto com o participante reforçando que o centro da avaliação é o produto e não o participante em si. O participante deve ser informado que ele estará sendo observado e fotografado e que a integridade do participante será totalmente resguardada, sendo utilizada a observação e as imagens somente para fins de análise do teste. O avaliador deve reforçar outras informações constantes do script e retirar dúvidas do participante sobre a sessão de teste.
4. Após serem passadas as orientações, o avaliador informará ao participante que ele pode utilizar o sistema livremente durante cinco minutos.
5. Passado este tempo, o avaliador irá entregar a lista de tarefas para execução (Lista de Tarefas). Os acontecimentos observados pelo avaliador deverão ser registrados no formulário de Coleta de Dados pelo Avaliado.
6. Depois de completadas todas as tarefas, o avaliador irá entregar ao participante o Questionário de Avaliação do Sistema pelo Participante para ser completado.
7. Passada pausa para o café, terá início a sessão de questionamento do participante sendo usado como guia o formulário de Tópicos para Questionamento. Outros tópicos além dos descritos neste formulário deverão ser acrescentados de acordo com os acontecimentos ocorridos durante o teste.
8. O avaliador agradece ao participante e se despede.

## **B.7 Formulários Utilizados**

1. Roteiro do Avaliador;
2. Questionário para Identificação do Perfil do Participante;
3. Script de Orientação;
4. Lista de Tarefas;
5. Coleta de Dados pelo Avaliador;
6. Questionário de Avaliação do Sistema pelo Participante;
7. Tópicos para Questionamento.

## Anexo C

# Questionário para Identificação do Perfil do Participante do Teste de Usabilidade

O objetivo deste questionário é colher informações sobre o perfil do participante do teste de usabilidade a ser realizado para validação do *método Mandos de interação* com usuários em aplicativos embarcados em telefones móveis.

As informações fornecidas são vitais para o aprimoramento do sistema.

Nas questões de marcar, favor circular a letra correspondente à resposta. A não ser que esteja indicado, deverá ser marcada somente uma resposta por questão.

Por favor, leia com atenção as questões a seguir e em caso de dúvida, solicite esclarecimento com o avaliador.

### 1) Informações Pessoais

1. Qual é a sua idade? \_\_\_\_\_ anos.
2. Sexo: M. masculino. F. feminino.

### 2) Informações Educacionais

1. Qual é o seu grau de instrução?
  - a. 2º grau incompleto
  - b. 2º grau completo
  - c. 3º grau incompleto
  - d. 3º grau completo

Escreva o nome do curso que está fazendo ou que completou de acordo com o grau assinalado acima:

-----

### 3) Experiência Profissional

1. Qual é a sua profissão?

-----

2. Há quanto tempo se encontra nesta profissão?

- a. Menos de 1 ano
- b. Entre 1 ano a 2 anos
- c. Entre 2 anos a 4 anos
- d. Mais de 4 anos

#### 4) Experiência Computacional

1. Há quanto tempo você utiliza aplicativos em telefone móvel?

- a. Entre 1 ano a 2 anos
- b. Entre 2 anos a 3 anos
- c. Entre 3 anos a 4 anos
- d. Mais de 4 anos.

2. Em que local você utiliza aplicativos em telefone móvel? (Pode-se marcar mais de uma opção)

- a. Em casa
- b. No trabalho
- c. Na escola
- d. Outros, favor especificar:

-----

3. Em média, quantas horas por semana você utiliza aplicativos em telefone móveis?

- a. Menos de 2 horas
- b. Entre 2 a 5 horas
- c. Entre 5 a 10 horas
- d. Mais de 10 horas

4. Quais ferramentas abaixo você utiliza em suas atividades diárias? (Pode-se marcar mais de uma opção)

- a. Despertador
- b. Calendário
- c. Calculadora
- d. Jogos
- e. Outros, favor especificar algum:

-----

# Anexo D

## Script de Orientação

Olá, meu nome é Mauro Teófilo, sou estudante de mestrado em informática da Universidade Federal do Amazonas e iremos trabalhar juntos nesta sessão de teste.

Estaremos efetuando o teste de dois protótipos de implantação de um novo método de interação com o usuário, o que consiste no uso do framework Mandos na criação de aplicativos embarcados em telefones móveis, desenvolvido para a validação do citado método de interação.

O teste ocorrerá na sala em que estamos. Esta sala apresenta condições para você não sofrer interferências na realização dos testes. Você manuseará dois aplicativos instalados no telefone celular BenQ-Siemens SL75, uma calculadora e um despertador.

Utilize os aplicativos a serem manuseados no telefone de forma normal e tranqüila, como se estivesse usando um outro aplicativo.

É importante que você diga o que está pensando durante a execução das tarefas.

Você poderá fazer perguntas, mas eu não poderei respondê-las. Isto irá ocorrer porque nós necessitamos verificar como você irá trabalhar com os aplicativos de forma independente.

Faça o melhor e não se preocupe com os resultados. É o método de interação que está sendo avaliada e não você. O método está sendo testado e com certeza, necessitará de modificações ou pode ser um método não eficaz e você estará contribuindo para detectar quais são as modificações necessárias, assim como validar a eficiência do mesmo.

Eu me sentarei próximo a você para tomar algumas notas.

Registrarei alguns momentos do teste com o uso de uma câmera fotográfica.

Você irá também responder a alguns questionários. É importante que sejam utilizadas informações verdadeiras e sinceras no preenchimento dos mesmos.

O nosso objetivo é descobrir vantagens e desvantagens na utilização destes aplicativos de acordo com a sua perspectiva, portanto necessitamos saber exatamente o que você pensa.

Você pode decidir invalidar seus dados, desde que me comunique até o final do teste.

Neste caso, seus dados e resultados não constarão do processo de análise do teste.

Sua integridade será totalmente preservada.

Estimamos cerca de quinze minutos para a duração desta sessão de testes.

Você tem alguma pergunta?

Se não, indicarei agora como você deve proceder para entrar nos aplicativos e a serem testados no telefone celular, assim como identificar qual faz uso do framework, podendo assim identificar as suas diferenças.

Depois de aprendido como se inicializa os aplicativos, utilize-os livremente durante cinco minutos e esteja à vontade para fazer perguntas neste momento.

Agradecemos por sua colaboração.

# Anexo E

## Lista de Tarefas

Agora, você dará início aos testes.

Abaixo, nós temos tarefas que devem ser executadas por você utilizando o produto.

As tarefas devem ser executadas na ordem em que se encontram.

Você deve ler em voz alta cada tarefa antes de executá-la.

Lembre-se:

- Verbalize suas dúvidas, pois isto ajudará ao avaliador anotar a ocorrência e a razão de problemas.
- É o produto que está sendo avaliado e não você.

Os testes serão realizados em duas etapas, que por sua vez será dividido em duas sessões.

A primeira etapa as tarefas a serem realizadas será do aplicativo Despertador, a primeira sessão será realizada utilizando o aplicativo que não faz uso do método de interação Mandos, na segunda sessão as mesmas tarefas serão realizadas, agora utilizando o aplicativo que faz uso do framework Mandos.

Na segunda etapa serão realizadas as tarefas do aplicativo Calculadora, na primeira sessão o aplicativo que faz uso do método de interação Mandos será testado, na segunda sessão será utilizado o aplicativo que não faz uso do método de interação Mandos.

### E.1 Etapa I – Despertador

Após a realização de cada tarefa o aplicativo ao ser encerrado, o estado inicial dele deve ser ajustado: o alarme deve estar desativado, a hora configurada para 12:00 AM e a pausa automática desabilitada.

Tarefa 1 – Iniciar o despertador, verificar o status do despertador e sair;

Tarefa 2 – Iniciar o despertador, ativar o alarme e sair;

Tarefa 3 – Iniciar o despertador, desativar o alarme e sair;

Tarefa 4 – Iniciar o despertador, ativar o alarme, configurar hora (8:00 AM) e sair;

Tarefa 5 – Iniciar o despertador, ativar o alarme, configurar hora (7:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta e sexta) e sair;

Tarefa 6 – Iniciar o despertador, configurar hora (6:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta, sexta e sábado), definir melodia e sair;

Tarefa 7 – Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a mais) e sair;

Tarefa 8 – Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a menos), habilitar pausa automática e sair;

## **E.2 Etapa II – Calculadora**

Tarefa 1 – Iniciar a calculadora, somar 42 com 33, ver o resultado e sair;

Tarefa 2 – Iniciar a calculadora, multiplicar 56 com 22, ver o resultado, salvar o resultado na memória, somar 47 com 11, somar o resultado obtido com o armazenado na memória e sair;

Tarefa 3 – Iniciar a calculadora, abrir a memória, limpar a memória, somar 30 com 65 e entrar no modo conversor, voltar para calculadora e sair;

Tarefa 4 – Iniciar a calculadora, ir para o modo ampliado, tirar a raiz quadrada de 45, salvar na memória, ir para o modo básico, diminuir 32 de 14,5 e somar com o resultado armazenado na memória e sair;

Tarefa 5 – Iniciar a calculadora, realizar a operação  $20 / 45$ , ver o resultado, limpar a tela e sair;

## Anexo F

### Coleta de Dados pelo Avaliador

O objetivo deste documento é ser utilizado pelo avaliador para a coleta manual de informações originadas da observação do participante durante o teste do protótipo do Método de interação com o usuário em aplicativos embarcados em telefones móveis.

Data e hora de início do teste: \_\_\_/\_\_\_/\_\_\_ h \_\_\_ min

Data e hora de fim do teste: \_\_\_/\_\_\_/\_\_\_ h \_\_\_ min

Item	Instrução apresentado ao participante	Detalhamento da Tarefa TME: Tempo máximo para execução.	Detalhes observados	n de erros	Sucesso (S / N)	Observações acerca da execução da tarefa
1	Iniciar o despertador, verificar o status do despertador e sair;	Aplicativo sem o uso do framework Mandos. TME: 30 s				
2	Iniciar o despertador, ativar o alarme e sair;	Aplicativo sem o uso do framework Mandos. TME: 40 s				
3	Iniciar o despertador, desativar o alarme e sair;	Aplicativo sem o uso do framework Mandos. TME: 40 s				
4	Iniciar o despertador, ativar o alarme, configurar hora (8:00 AM) e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min				
5	Iniciar o despertador, ativar o alarme, configurar hora (7:00 AM), configurar Dias (habilitar segunda, terça,	Aplicativo sem o uso do framework Mandos. TME: 1 min				

	quarta, quinta e sexta) e sair;					
6	Iniciar o despertador, configurar hora (6:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta, sexta e sábado), definir melodia e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min e 30s				
7	Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a mais) e sair;	Aplicativo sem o uso do framework Mandos. TME: 2 min				
8	Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a menos), habilitar pausa automática e sair;	Aplicativo com o uso do framework Mandos. TME: 2 min				
9	Iniciar o despertador, verificar o status do despertador e sair;	Aplicativo com o uso do framework Mandos. TME: 30 s				
10	Iniciar o despertador, ativar o alarme e sair;	Aplicativo com o uso do framework Mandos. TME: 40 s				
11	Iniciar o despertador, desativar o alarme e sair;	Aplicativo com o uso do framework Mandos. TME: 40 s				
12	Iniciar o despertador, ativar o alarme, configurar hora (8:00 AM) e sair;	Aplicativo com o uso do framework Mandos. TME: 1 min				
13	Iniciar o despertador, ativar o alarme, configurar hora (7:00 AM), configurar	Aplicativo com o uso do framework Mandos. TME: 1 min				

	Dias (habilitar segunda, terça, quarta, quinta e sexta) e sair;					
14	Iniciar o despertador, configurar hora (6:00 AM), configurar dias (habilitar segunda, terça, quarta, quinta, sexta e sábado), definir melodia e sair;	Aplicativo com o uso do framework Mandos. TME: 1 min e 30s				
15	Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a mais) e sair;	Aplicativo com o uso do framework Mandos. TME: 2 min				
16	Iniciar o despertador, configurar hora (7:00AM), definir melodia, ajustar relógio do celular (para 5 minutos a menos), habilitar pausa automática e sair;	Aplicativo com o uso do framework Mandos. TME: 2 min				
17	Iniciar a calculadora, somar 42 com 33, ver o resultado e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min e 30s				
18	Iniciar a calculadora, mutiplicar 56 com 22, ver o resultado, salvar o resultado na memória, somar 47 com 11, somar o resultado obtido com o armazenado na memória e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min e 30s				
19	Iniciar a calculadora, abrir a memória, limpar a memória, somar 30 com 65 e entrar no modo conversor, voltar para calculadora e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min e 30s				
20	Iniciar a calculadora, ir para o modo	Aplicativo sem o uso do framework				

	aplicado, tirar a raiz quadrada de 45, salvar na memória, ir para o modo básico, dominuir 32 de 14,5 e somar com o resultado armazenado na memória;	Mandos. TME: 1 min e 30s				
21	Iniciar a calculadora, realizar a operação 20 / 45, ver o resultado, limpar a tela;	Aplicativo sem o uso do framework Mandos. TME: 1 min				
22	Iniciar a calculadora, somar 42 com 33, ver o resultado e sair;	Aplicativo sem o uso do framework Mandos. TME: 1 min e 30s				
23	Iniciar a calculadora, mutiplicar 56 com 22, ver o resultado, salvar o resultado na memória, somar 47 com 11, somar o resultado obtido com o armazenado na memória e sair;	Aplicativo com o uso do framework Mandos. TME: 1 min e 30s				
24	Iniciar a calculadora, abrir a memória, limpar a memória, somar 30 com 65 e entrar no modo conversor, voltar para calculadora e sair;	Aplicativo com o uso do framework Mandos. TME: 1 min e 30s				
25	Iniciar a calculadora, ir para o modo aplicado, tirar a raiz quadrada de 45, salvar na memória, ir para o modo básico, dominuir 32 de 14,5 e somar com o resultado armazenado na memória;	Aplicativo com o uso do framework Mandos. TME: 1 min e 30s				
26	Iniciar a calculadora, realizar a operação 20 / 45, ver o resultado, limpar a tela;	Aplicativo com o uso do framework Mandos. TME: 1 min				

## Anexo G

# Questionário de Avaliação do Sistema pelo Participante

O objetivo deste questionário é colher informações sobre a opinião do participante do teste de usabilidade que foi realizado utilizando o protótipo do Método de Interação com o usuário em aplicativos embarcados em telefones móveis.

As informações fornecidas são vitais para o aprimoramento e validação do sistema.

Nas questões de marcar, favor circular o número correspondente ao grau de concordância. A não ser que esteja indicado, deverá ser marcada somente uma resposta por questão.

Por favor, leia com atenção as questões a seguir e em caso de dúvida, solicite esclarecimento com o avaliador.

1. Favor marcar o número correspondente ao grau que você mais concorda:

**Intuitividade** - Entre os atributos que compõe a usabilidade, o mais importante é a intuitividade. Para que ocorra minimamente uma interação, a interface deve apresentar características que facilitem sua utilização permitindo que usuários básicos ou avançados possam aprender seus recursos de forma clara e objetiva.

Aplicativos	Avaliação				
<b>Sem</b> o uso do método Mandos	1	2	3	4	5
<b>Com</b> o uso do método Mandos	1	2	3	4	5

**Eficiência** - A eficiência é determinada através de quanto tempo é gasto utilização do software para realização de uma tarefa. Quanto menor o tempo gasto, ou menor o esforço despendido, mais eficiente é o aplicativo.

Aplicativos	Avaliação				
<b>Sem</b> o uso do método Mandos	1	2	3	4	5
<b>Com</b> o uso do método Mandos	1	2	3	4	5

**Memorização** - Denota o quanto o aplicativo é fácil de ser memorizado, ou seja, após a aprendizagem de uma tarefa, se o aplicativo auxilia de forma a tornar a realização desta tarefa mais fácil de ser memorizada.

Aplicativos	Avaliação				
<b>Sem</b> o uso do método Mandos	1	2	3	4	5
<b>Com</b> o uso do método Mandos	1	2	3	4	5

**Erros** - É considerado erro qualquer ação que leve o usuário a não executar determinada tarefa não se levando em conta os possíveis impactos causados por diferentes tipos de erros. Um bom aplicativo apresenta uma interação com o usuário de forma que o mesmo tenha menor chance de cometer erros em uma tarefa.

Aplicativos	Avaliação				
<b>Sem</b> o uso do método Mandos	1	2	3	4	5
<b>Com</b> o uso do método Mandos	1	2	3	4	5

**Satisfação** - A satisfação representa o quão agradável deve ser a interação do usuário com o sistema.

Aplicativos	Avaliação				
<b>Sem</b> o uso do método Mandos	1	2	3	4	5
<b>Com</b> o uso do método Mandos	1	2	3	4	5

2. Aponte situações em que você achou mais fácil utilizar o sistema que faz uso do método Mandos e também o que achou mais fácil sem o uso do método:

-----

-----

-----

-----

3. Aponte situações em que você achou mais difícil utilizar o sistema que faz uso do método Mandos e também sobre o que achou mais difícil sem o uso do método:

-----

-----

-----

-----

4. Qual a principal diferença entre os dois métodos?

-----

-----

-----

-----

5. Qual a sua avaliação sobre o método de interação Mandos?

-----

-----

-----

-----

6. O espaço abaixo é reservado para que você exponha sua opinião e sugira melhorias no sistema.

-----

-----

-----

-----

-----

-----

## Anexo H

### Tópicos para Questionamento

O objetivo deste questionário é sugerir alguns tópicos a serem discutidos em uma sessão de questionamento do participante após a realização do teste do método de interação com o usuário em aplicativos embarcados em telefones móveis.

Discussões a serem levantadas:

1. Você se sentiria satisfeito se o método de interação Mandos fosse utilizado no seu celular?

2. Se fosse possível você próprio editar a interface com o usuário através de um programa, escolhendo as tarefas mais usuais e habilitando isto, você teria interesse em utilizar?

3. Você se sentiu confuso em algum momento com o uso do método Mandos?