



**Universidade Federal do Ceará
Centro de Ciências
Departamento de Computação
Mestrado e Doutorado em Ciência da Computação**

Dissertação de Mestrado

Requisitos para Testes de Aplicações Móveis

Valéria Lelli Leitão Dantas

Fortaleza – Ceará
2009

Valéria Lelli Leitão Dantas

Requisitos para Testes de Aplicações Móveis

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-graduação em Ciência da Computação (MDCC) da Universidade Federal do Ceará (UFC) como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Rossana Maria de Castro Andrade, PhD.

Requisitos para Testes de Aplicações Móveis

Dissertação de Mestrado submetida à Coordenação do Programa de Pós-graduação em Ciência da Computação (MDCC) da Universidade Federal do Ceará (UFC) como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

Aprovado em ____/____/____

Banca Examinadora

Profa. Rossana Maria de Castro Andrade, PhD. (Orientadora)
Universidade Federal do Ceará – UFC

Prof. Auri Marcelo Rizzo Vincenzi, DSc.
Universidade Federal de Goiás – UFG

Prof. Pedro de Alcântara dos Santos Neto, DSc.
Universidade Federal do Piauí – UFPI

Prof. José Antônio Fernandes de Macêdo, DSc.
Universidade Federal do Ceará – UFC

Dedico esta dissertação

Aos meus pais Ademir e Cida pelo amor e
carinho.

Aos meus irmãos Marcelo, Cláudia e Priscila
porque serão sempre parte da minha vida.

Ao meu esposo Manoel, pelo amor,
companheirismo, incentivo e paciência.

Agradecimentos

Agradeço à Deus por me dar forças e compreender melhor a vida.

Aos meus pais, Ademir e Cida, pelo exemplo de caráter e pelos incentivos nos estudos. Vocês são os meus maiores tesouros.

Ao meu amado esposo, Manoel, por todo amor, companheirismo e acima de tudo paciência nos momentos mais difíceis.

Aos meus irmãos, Marcelo e Priscila pelas nossas conversas e por sempre serem parte da minha vida. À Claudía, em especial, por ter nos presenteado com os mais lindos sobrinhos, João Henrique, Arthur e a pequena Julinha.

À minha orientadora, a professora Rossana, pela idéia inicial do trabalho, pelo incentivo, por confiar e acreditar na minha capacidade. Aos servidores do GREat, por todo suporte necessário para a realização das minhas atividades, em especial, à Christiane, Danilo, Darilu, Fernando, Liliane, Sr. Adir, Ronaldo e Suzana.

Por último, obrigado a todos os amigos de pesquisa e projetos no GREat, em especial (ordem alfabética) Alexsandro, Aline, Bosco, Bruno Góis, Bruno Sabóia, Clayson, Carina, Carlos Pontual, Cláudio, Davi, Davyd, Dário, Diana, Fabiana, Fabrício, Fátima, Flávio, Jefferson, Joaquim, João Borges, João Marcelo, Levi, Lincoln, Luana, Luís, Márcio, Mirko, Neto, Reinaldo, Ricardo, Ronaldo Mota, Rute, Saulo, Smaylle, Tales e Zacarias.

Obrigado ainda à FUNCAP, que deu suporte financeiro indispensável à conclusão desse mestrado.

“...o ponto alto da vida de um homem, a sua maior realização, é aquele momento em que ele deu sangue por uma boa causa e está caído, exausto, no campo de batalha – vitorioso.”

Vince Lombardi

Resumo

O avanço tecnológico das redes sem fio contribuiu para tornar os dispositivos móveis cada vez mais sofisticados e acessíveis para o desenvolvimento de aplicações. Para atender essa crescente demanda surge a necessidade de desenvolver sistemas confiáveis, corretos e seguros. Nesse contexto, os testes de software são um dos instrumentos para suprir essa necessidade. Entretanto, no desenvolvimento de aplicações para dispositivos móveis, o processo de teste além de ser planejado, tem que ser realizado em função da mobilidade e de outras limitações do ambiente onde está inserido, denominado de ambiente móvel nesta dissertação. Sendo assim, mobilidade e limitações dos dispositivos, tais como pouca memória, baixo processamento, tela pequena, capacidade de armazenamento e consumo de energia, demandam uma adaptação no processo tradicional de teste de software. Com o intuito de tratar as questões intrínsecas do ambiente móvel, esta dissertação propõe requisitos para o teste de aplicações móveis, os quais são classificados em requisitos para o processo de teste de aplicações móveis e requisitos para o teste de usabilidade de aplicações móveis. Um estudo de caso é realizado para avaliar a utilização desses requisitos no teste de aplicações móveis e demonstrar que o seu uso identifica problemas específicos do ambiente móvel e, portanto, propicia uma melhoria no processo de teste dessas aplicações.

Palavras-chave: Computação Móvel, Teste de Software e Qualidade de Software.

Abstract

Wireless networks have contributed to the technological advance that has popularized the use of mobile devices and fostered the development of applications targeted at these devices. This growing demand has motivated the research to develop reliable, accurate and secure systems. In this context, software testing is one of the tools to achieve this goal. However, when developing applications for mobile devices, the testing process should be planned and executed considering device mobility and other constraints of the mobile environment where this device is inserted. Device limitations, such as low processing, memory, storage and battery, as well as small screen sizes, require an adjustment in the traditional software testing process. In order to address the issues inherent to the mobile environment, this master thesis proposes a set of requirements for testing mobile applications. They are divided into two categories, namely testing process requirements for mobile applications and usability testing requirements for mobile applications. A case study is conducted to demonstrate that the use of these requirements identifies specific problems of the mobile environment, improving the testing process for mobile applications.

Key-words: Mobile Computing, Software Testing, and Software Quality.

Sumário

LISTA DE FIGURAS	11
LISTA DE TABELAS	12
LISTA DE ABREVIATURAS	13
CAPÍTULO 1.....	14
INTRODUÇÃO	14
1.1 CONTEXTUALIZAÇÃO E CARACTERIZAÇÃO DO PROBLEMA	14
1.2 MOTIVAÇÃO	16
1.3 OBJETIVOS E CONTRIBUIÇÕES	18
1.4 ORGANIZAÇÃO DA DISSERTAÇÃO.....	19
CAPÍTULO 2.....	20
AMBIENTE MÓVEL.....	20
2.1 MOBILIDADE	20
2.2 CARACTERÍSTICAS DO AMBIENTE MÓVEL	20
2.2.1 Contexto Móvel	21
2.2.1.1 Dispositivos Móveis e Desktop	25
2.2.2 Usuário Móvel	29
2.2.3 Aplicação Móvel	30
2.2.3.1 Tipos de Aplicações Móveis.....	32
2.3 CONCLUSÃO	33
CAPÍTULO 3.....	34
TESTES DE SOFTWARE.....	34
3.1 VERIFICAÇÃO E VALIDAÇÃO.....	34
3.2 CONCEITOS DE TESTES DE SOFTWARE	35
3.2.1 Casos de Teste.....	37
3.2.2 Tipos de Teste	38
3.3 TESTES PARA APLICAÇÕES MÓVEIS	39
3.4 Usabilidade e experiência do usuário.....	43
3.5 Teste de usabilidade nas aplicações móveis	44
3.6 Conclusão	46
CAPÍTULO 4.....	48
REQUISITOS PARA TESTES	48
4.1 METODOLOGIA	48
4.1.1 Levantamento teórico-prático	49
4.1.2 Coleta de dados.....	49
4.1.3 Análise dos dados	51
4.1.4 Validação	51
4.2 REQUISITOS PROPOSTOS.....	51
4.2.1 Requisitos para o Processo de Teste de Aplicações Móveis	52
4.2.2 Requisitos para o Teste de Usabilidade de Aplicações Móveis	61
4.3 RESUMO DOS REQUISITOS PROPOSTOS	66
4.4 CONCLUSÕES	68
CAPÍTULO 5.....	70
VALIDAÇÃO	70
5.1 ESTUDO DE CASO	70
5.2 AMBIENTE DE TESTE	71
5.2.1 Shopping Móvel, Museu Móvel e Teatro Móvel.....	72
5.2.2 Outras Aplicações.....	74
5.3 RESULTADOS DA UTILIZAÇÃO DOS REQUISITOS PROPOSTOS	75
5.3.1 Aplicação Shopping Móvel.....	75

5.3.2	<i>Aplicação Museu Móvel</i>	86
5.3.3	<i>Aplicação Teatro Móvel</i>	96
5.3.4	<i>Outras Aplicações</i>	104
5.4	DISCUSSÃO DOS RESULTADOS ALCANÇADOS.....	110
5.5	CONCLUSÃO	115
CAPÍTULO 6.....		116
CONCLUSÃO E TRABALHOS FUTUROS.....		116
6.1	CONTRIBUIÇÕES	116
6.2	TRABALHOS FUTUROS.....	118
REFERÊNCIAS BIBLIOGRÁFICAS		120

Lista de Figuras

Figura 1.1 - Processo de Desenvolvimento e Garantia de Qualidade para Aplicações Móveis. Adaptada de [Nokia Forum 2009].....	17
Figura 2.1 – A continuação do computador pessoal. Adaptada de [Weiss 2002].....	22
Figura 2.2 – Características PCD. Adaptada de [Ballard 2007]	24
Figura 2.3 – Classificação dos dispositivos móveis em relação ao tamanho da tela	25
Figura 4.1 - Tipos de aplicações móveis.....	50
Figura 4.2– Perfil dos entrevistados.....	50
Figura 4.3 - Informações sobre o erro reportado	56
Figura 4.4 – Variedade de modelos de dispositivos em relação ao <i>design</i>	65
Figura 4.5 - Tipos de teste realizados na prática	68
Figura 5.1 – Imagem com resolução original 800*600 pixels visualizada no N95	78
Figura 5.2 – Shopping Móvel: Visualização de mensagem de erro no WTK	80
Figura 5.3 - Visualização de mensagem de erro no JP8 e no G705.....	81
Figura 5.4 – Visualização de mensagem de erro no S60 e no N95.....	81
Figura 5.5 – Diferenças de visualização de texto no S60 e no N95.....	83
Figura 5.6 – Nome da loja alinhado à direita no G705.....	84
Figura 5.7 – Menu de opções “Selecionar” na <i>softkey</i> da esquerda no G705.....	86
Figura 5.8 – Telas sobrepostas do Museu Móvel no G705	88
Figura 5.9 – Imagens visualizadas no JP8 e no G705.....	89
Figura 5.10 – Imagens visualizadas no S60 e no N95.....	89
Figura 5.11 – Deficiência no foco de seleção no G705.....	92
Figura 5.12 – Museu Móvel: Visualização do Texto no WTK, no S60 e no N95.....	94
Figura 5.13 – Visualização do Texto no JP8 e no G705	95
Figura 5.14 – Museu Móvel: rótulos das <i>softkeys</i> no G705	96
Figura 5.15 – Mensagem de erro	99
Figura 5.16 – Lista das peças.....	99
Figura 5.17 – Tela de reservas.....	100
Figura 5.18 – Tela inicial do Teatro Móvel.....	100
Figura 5.19– Resumo da peça visualizado respectivamente nos formatos retrato e paisagem	101
Figura 5.20– Visualização do texto no JP8 e no G705	102
Figura 5.21 – Visualização do texto no JP8 e no G705.....	103

Lista de Tabelas

Tabela 2.1 – Comparação do dispositivo móvel e computador de mesa. Adaptada de [Weiss 2002]	29
Tabela 4.1 – Características testadas no emulador e no dispositivo móvel.....	54
Tabela 4.2 – Limitações testadas na prática nas aplicações para dispositivos móveis	58
Tabela 4.3 – Características do Teste em Laboratório e em Campo	60
Tabela 4.4– Requisitos para o Processo de Teste de Aplicações Móveis	66
Tabela 4.5 – Requisitos para o Teste de Usabilidade de Aplicações Móveis	67
Tabela 5.1 – Especificação dos celulares utilizados nos testes	73
Tabela 5.2 – Características do projeto das aplicações móveis testadas.....	75
Tabela 5.3 – Quantidade de erros encontrados em cada ambiente na aplicação Shopping Móvel	76
Tabela 5.4 – Resultados dos testes de acordo com as limitações testadas no Shopping Móvel	79
Tabela 5.5 – Quantidade de erros encontrados em cada ambiente do Museu Móvel	87
Tabela 5.6 – Resultados dos testes de acordo com as limitações testadas no Museu Móvel	90
Tabela 5.7 – Quantidade de erros encontrados em cada ambiente no Teatro Móvel	96
Tabela 5.8– Resultados dos testes de acordo com as limitações testadas do Teatro Móvel.....	98
Tabela 5.9– Características do projeto de desenvolvimento das aplicações móveis	105
Tabela 5.10– Características testadas nos Emuladores Genéricos, Específicos e nos Celulares	112
Tabela 5.11– Características testadas em laboratório.....	113

Lista de Abreviaturas

BREW	Binary Runtime Environment
BMP	Bitmap
CSK	Center Soft Key
DVD	Digital Video Disc
ES	Engenharia de Software
GB	Gigabyte
GPRS	General Packet Radio Service
GPS	Global Positioning System
GSM	Global System for Mobile communications
IrDA	Infrared Data Association
JPG	Joint Photographic Group
JME	Java Micro Edition
KB	Kilobyte
KVM	Kilo Virtual Machine
MB	Megabyte
MMS	Multimedia Messaging Service
PC	Personal Computer
PCD	Personal Communications Device
PNG	<i>Portable Network Graphics</i>
PDA	Personal Digital Assistant
RAM	Random Access Memory
RFID	Radio-Frequency IDentification
RSK	Right Soft Key
SQA	Software Quality Assurance
SMS	Short Message Service
UI	User Interface
UFC	Universidade Federal do Ceará
V&V	Verificação e Validação
V&VT	Verificação, Validação e Testes
WAP	<i>Wireless Application Protocol</i>
Wi-Fi	Wireless Fidelity
WTK	Wireless Toolkit
WWW	World Wide Web
XP	Extreme Programming

Capítulo 1

Introdução

Esta dissertação propõe requisitos que tem por objetivo melhorar o processo de teste e usabilidade das aplicações para dispositivos móveis. Este capítulo introduz, na Seção 1.1, uma visão geral sobre o processo de teste de aplicações para dispositivos móveis e seus desafios. Em seguida, a Seção 1.2 expõe a motivação para o desenvolvimento deste trabalho. Na Seção 1.3 são descritos os principais objetivos desta dissertação. A Seção 1.4 apresenta a metodologia utilizada para a realização deste trabalho e, por fim, a Seção 1.5 apresenta a estrutura organizacional desta dissertação.

1.1 Contextualização e Caracterização do Problema

O grande avanço tecnológico das redes sem fio popularizou o uso dos dispositivos móveis (e.g., PDAs – Personal Digital Assistant, smartphones e telefone celulares). A grande variedade de modelos no mercado resulta na demanda de novos aplicativos e serviços que denominados nesta dissertação de aplicações móveis. Estas são desenvolvidas para executar em dispositivos móveis e interagem com o usuário, que pode estar em movimento, através da tela do dispositivo móvel. Esse usuário será referenciado nesta dissertação como usuário móvel.

As aplicações móveis são complexas devido à mobilidade que é proporcionada pelas tecnologias sem fio. De acordo com [Lee et al. 2005], a mobilidade é a capacidade do usuário de se deslocar facilmente. No ambiente móvel, a mobilidade se refere ao uso, por usuários, de dispositivos móveis portáteis que ofereçam um conjunto de serviços, e envolve questões tais como portabilidade, usabilidade, funcionalidade e conectividade. Em virtude disso, o desenvolvimento de aplicações móveis se torna complexo e as mesmas nem sempre vão de encontro às expectativas dos usuários [Zeidler et al. 2007].

A complexidade para desenvolver as aplicações móveis e, principalmente, testá-las sob um ambiente móvel exige uma adaptação no processo de teste tradicional. Dessa forma, desenvolver e testar aplicações móveis com qualidade tornou-se um dos desafios a ser solucionado pelos engenheiros de software.

De acordo com [Delamaro et al. 2007] para atingir qualidade de software é necessário disponibilizar sistemas confiáveis, corretos e seguros. Contudo, tal dimensão de qualidade só

pode ser alcançada quando o teste é executado como um processo próprio [Bastos et al. 2007]. Assim, a etapa de testes não vem sendo mais considerada apenas como uma etapa do processo de desenvolvimento das aplicações tradicionais, que são aplicações desenvolvidas para executar em um computador de mesa (ambiente *desktop*), mas sim como um processo à parte que deve ser executado em paralelo ao desenvolvimento e iniciado logo que os requisitos estejam definidos. Neste trabalho, consideraremos essa visão ao propor requisitos para o processo de teste de aplicações móveis.

A principal diferença das aplicações desenvolvidas em um ambiente *desktop* (aplicações *desktop*) para as aplicações móveis tratadas nesta dissertação é a mobilidade e as limitações proporcionadas pelo ambiente, aqui denominado de ambiente móvel. Este envolve aplicações móveis, usuário móvel e contexto móvel.

Dessa forma, o ambiente móvel muda a forma “estática” de se testar as aplicações. Os testes das aplicações *desktop* são executados no computador fixo sobre a mesa dentro de um laboratório com o testador e/ou usuário sentado usando uma aplicação *desktop*.

Por outro lado, o ambiente de teste das aplicações móveis é mais dinâmico porque as questões relacionadas à mobilidade devem ser acomodadas ao processo bem como as limitações do ambiente *desktop* e as limitações do ambiente móvel. Além disso, o usuário móvel é passível de interrupção o que faz com que o ambiente ao seu redor possa distrair sua atenção interferindo na sua interação com a aplicação móvel. Em virtude disso, existe uma dificuldade do processo de teste utilizar uma metodologia que possa capturar todas as possibilidades do ambiente móvel em um simples teste.

Além disso, os testadores do ambiente *desktop*, por exemplo, não precisam se preocupar, com restrições de processamento, tela e consumo de energia. Se uma aplicação *desktop* durante os seus testes mostrou baixo desempenho em relação ao processamento, provavelmente não é devido às restrições de capacidade do processador do computador *desktop*.

Já no ambiente móvel, além da mobilidade, as limitações dos dispositivos, tais como memória, processamento, tela, bateria, capacidade de armazenamento e largura de banda impactam no processo de desenvolvimento das aplicações móveis.

Outro fator observado é que no teste de aplicações móveis, os testadores enfrentam problemas exclusivos do contexto móvel, dentre os quais temos:

- (i) os tipos de testes que devem ser feitos para cobrir as limitações desse ambiente;
- (ii) teste em diferentes tipos de aplicações móveis (e.g., corporativa, e-commerce, jogo e site) e na variedade de modelos de dispositivos;
- (iii) o que deve ser testado em laboratório e em campo e como fazê-lo;

- (iv) problemas relacionados a usabilidade decorrentes da capacidade do dispositivo (e.g., tamanho da tela e do teclado);

Nesse cenário, pode-se afirmar que o processo de teste tradicional falha em questões que são específicas do ambiente móvel e, portanto, deve ser adaptado para incorporar as restrições intrínsecas do ambiente móvel, além de ter que considerar as características comuns do processo de teste de aplicações tradicionais.

1.2 Motivação

Na literatura são encontrados diversos trabalhos que documentam processos de desenvolvimento específicos para aplicações móveis. Destacam-se nesta dissertação três processos que embora sejam para aplicações móveis possuem diferentes abordagens. O primeiro é o Processo de Desenvolvimento e Garantia de Qualidade para Aplicações Móveis que não depende do tipo de aplicação apresentado em [Nokia Forum 2009]; O segundo é o Processo de Desenvolvimento de Jogos Móveis que depende do tipo de conforme pode ser visto em [Morales e Nelson 2007] e o último é o Processo de Desenvolvimento do Produto Integrado que foca no produto apresentado em [Zeidler et al. 2007].

O Processo de Desenvolvimento e Garantia de Qualidade para Aplicações Móveis é ilustrado na Figura 1.1. Ele é composto de oito fases principais que inclui a fase de teste como parte do ciclo de desenvolvimento. Nesta fase recomenda-se o teste de sistema que é instalar a aplicação e executar os casos de testes em ambiente de tempo de execução real no dispositivo alvo ou outros dispositivos móveis. A execução é feita de acordo com o plano de teste que especifica o programa, o escopo e o conteúdo da fase de teste incluindo os casos de testes. Os erros são reportados em um relatório que lista e descreve todos os erros encontrados durante o teste.

O processo de desenvolvimento de jogos móveis é proposto de acordo com o processo de *design* de jogos. Dessa forma, o processo funciona mais como um guia de criação de jogos móveis do que um processo de desenvolvimento propriamente dito. Ele foca em questões de usabilidade que devem ser pensadas durante o desenvolvimento, tais como limitada resolução da tela, controles de entrada variável, poder de processamento variável, conexão da rede e ambiente ao redor dos jogadores. Os testes não são abordados, no entanto, do ponto de vista do autor como os jogos móveis rodam em diferentes modelos de celulares, o maior problema é o desenvolvedor ter que fazer diferentes versões do jogo para celulares específicos alvo e dessa forma, gastar mais tempo nos testes do que o normal.

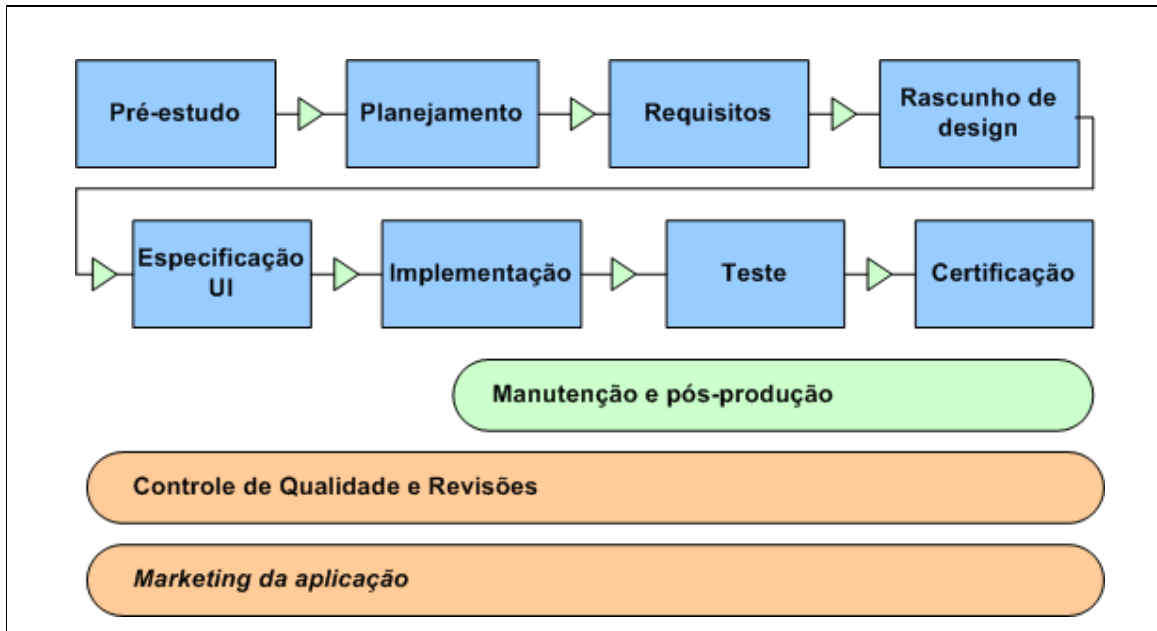


Figura 1.1 - Processo de Desenvolvimento e Garantia de Qualidade para Aplicações Móveis.
Adaptada de [Nokia Forum 2009]

O processo de desenvolvimento do produto integrado é composto de cinco passos, os quais são: 1) geração e avaliação de idéias; 2) desenvolvimento de modelo de negócios e caso de negócios; 3) aspectos legais; 4) projeto de pesquisa de mercado e experiência do usuário e 5) Implementação. Os testes são realizados na fase de implementação onde o software é construído. Uma lista com os dispositivos móveis relevantes para o mercado alvo é elaborada desde cedo no projeto. Dessa forma, os testes são feitos diretamente nos dispositivos móveis alvo e conduzidos por planos de teste padronizados e protocolos de testes.

Além dos processos de desenvolvimento mencionados acima, foi encontrado na literatura um processo de teste para desenvolvimento de jogos celulares, conforme apresentado em [Fernandes e Dias 2006]. O processo de teste é apresentado de acordo com as etapas de desenvolvimento definidas no próprio trabalho (e.g., alfa, beta, *candidate* e *gold*). Para que cada versão desenvolvida do jogo passe para a etapa seguinte de desenvolvimento são realizados testes específicos. Estes são classificados em quatro fases: testes funcionais, testes de certificação, testes de plataforma de comunicação do jogo e teste de publicação (do inglês *publishing*).

Os testes funcionais garantem que o jogo contenha todos os requisitos implementados. Os testes de certificação preocupam-se com os padrões internacionais de qualidade para jogos celulares. Os testes de plataforma de comunicação do jogo validam a comunicação do jogo com algum servidor computacional e os testes de publicação verificam se o jogo é executado no aparelho celular

No entanto, embora o processo de teste proposto seja específico para jogos celulares e aborde fatores comuns no desenvolvimento de jogos tradicionais, ele não trata as questões inerentes do ambiente móvel que podem afetar o uso da aplicação. Por exemplo, limitações dos dispositivos, tais como pouca memória e baixo processamento podem afetar o desempenho do jogo. Além disso, a mobilidade do usuário móvel pode interferir na jogabilidade do jogo.

De acordo com os modelos de processo apresentados pode-se afirmar que na prática os testes das aplicações móveis são deficientes por não considerar questões de mobilidade e do ambiente móvel que são fundamentais para o processo de teste de aplicações móveis.

Dessa forma, avaliar as aplicações móveis de forma adequada de modo a proporcionar uma melhoria na qualidade através dos testes requer dos especialistas um conhecimento aprofundado da tecnologia móvel e das limitações impostas por esta. Porém, na prática, esta tarefa não é trivial e requer mudanças no processo de teste tradicional.

Devido a essa complexidade, neste trabalho, são propostos requisitos para o teste de aplicações móveis que tratam questões intrínsecas do ambiente móvel. Esses requisitos devem ser adicionados ao processo de teste com o intuito de minimizar os desafios encontrados no ambiente móvel e conduzir os testadores durante o processo para o mesmo se torne mais eficaz.

1.3 Objetivos e Contribuições

Este trabalho tem como objetivo propor requisitos específicos para o teste de aplicações móveis, os quais são elicitados a partir de extensa revisão bibliográfica e pesquisa com profissionais experientes no desenvolvimento e testes da área de computação móvel.

Para atingir o objetivo deste trabalho as seguintes atividades são previstas:

- Investigar o processo de desenvolvimento das aplicações móveis;
- Fazer um levantamento de como as aplicações móveis são testadas;
- Elicitar os requisitos para o teste de aplicações móveis, classificá-los de acordo com as suas finalidades e propô-los;
- Validar os requisitos elicitados em três aplicações móveis desenvolvidas pelos alunos da disciplina de Engenharia de Software (ES) em 2006 e aplicá-los na disciplina de ES em 2009 durante o projeto de desenvolvimento de aplicações móveis;

Os requisitos propostos neste trabalho ajudam os testadores a encontrar problemas específicos do ambiente móvel e, dessa forma, acredita-se que isso possa melhorar a

produtividade e eficácia dos testes das aplicações móveis tanto no processo de teste propriamente dito quanto na usabilidade.

1.4 Organização da dissertação

Além deste capítulo introdutório, esta dissertação está estruturada em mais cinco capítulos e um anexo descritos a seguir.

Capítulo 2 – Ambiente Móvel: tem por objetivo fornecer o embasamento teórico acerca do ambiente móvel para um melhor entendimento da complexidade dos testes em aplicações móveis. Dessa forma, são apresentadas as características que envolvem o ambiente móvel: o contexto móvel, a aplicação móvel e o usuário móvel.

Capítulo 3 – Testes de Software: apresenta uma visão geral sobre teste de software e os principais conceitos relacionados com a área. Este capítulo aborda principalmente, o teste de aplicações móveis e questões de usabilidade relacionadas com a área.

Capítulo 4 – Requisitos para o Teste: apresenta os requisitos para o teste de aplicações móveis classificados de acordo com o processo de teste e a usabilidade. Também apresenta a metodologia utilizada para a elicitacão dos requisitos e os resultados obtidos a partir das entrevistas e dos questionários.

Capítulo 5 – Validação: apresenta a análise dos resultados obtidos com a utilização dos requisitos elicitados nas três aplicações móveis desenvolvidas pelos alunos de ES em 2006 e durante a disciplina de ES em 2009.

Capítulo 6 – Considerações Finais e Trabalhos Futuros: apresenta as conclusões sobre este trabalho e seus futuros direcionamentos.

Anexo A – Casos de testes das aplicações móveis: este anexo apresenta os casos de testes das três aplicações móveis desenvolvidas pelos alunos da disciplina de ES em 2006.

Nesta dissertação são utilizadas algumas notações que visam facilitar a leitura:

- Os termos em *itálico* são usados para representar palavras ou expressões em língua estrangeira.
- Os termos em **negrito** são utilizados para destacar os requisitos elicitados que serão apresentados ao longo do texto.
- Notas de rodapé são utilizadas para fazer algum comentário relevante ou alguma explicação adicional sobre termos utilizados no texto.

Capítulo 2

Ambiente Móvel

O objetivo deste capítulo é fornecer o embasamento teórico sobre o ambiente móvel que é considerado nesta dissertação. A Seção 2.1 apresenta mobilidade como fator chave de um ambiente móvel. A Seção 2.2 apresenta as características do ambiente móvel, sendo que na subseção 2.2.1 é discutido as principais características do contexto móvel, na subseção 2.2.2 são apresentadas as características do usuário móvel, na subseção 2.2.3 são identificadas as características e os tipos de aplicações móveis e, finalmente, a Seção 2.3 traz as principais considerações sobre este capítulo.

2.1 Mobilidade

A mobilidade é o fator intrínseco do ambiente móvel onde a qualquer momento e em qualquer lugar o acesso as informações e aos serviços é possível. No ambiente móvel, mobilidade se refere ao uso pelos usuários de dispositivos móveis portáteis que ofereçam um conjunto de serviços. Dessa forma, novas funcionalidades são acopladas ao dispositivo móvel, tais como texto, voz, navegação *web*, câmera, música e televisão.

A introdução de novas funcionalidades e a melhoria dos recursos dos dispositivos aumentou a demanda por aplicações móveis e isso tem estimulado o seu uso em diversas áreas, tais como comercial, entretenimento, saúde, transporte, assistência técnica, turismo cultural, dentre outras.

No entanto, o termo “móvel” nem sempre é empregado da mesma forma. De acordo com Ballard [Ballard 2007], um *laptop* também é considerado um dispositivo móvel, mas que possui mecanismos de entrada e saída de dados semelhante a um computador *desktop* e, muitas vezes, é utilizado pelo usuário como tal, fixo sobre a mesa. Baseado nessa idéia, o termo móvel se refere à capacidade que o usuário tem de se deslocar e não ao dispositivo ou aplicações.

2.2 Características do Ambiente Móvel

Um ambiente móvel introduz limitações associadas com a diversidade dos dispositivos, a diversidade de plataformas de desenvolvimento (e.g., Brew, Palm, Windows Mobile, Symbian, e

Java Micro Edition – JME), a modalidade da aplicação (comunicação de voz e dados), os mecanismos de transferência de dados e a mobilidade do usuário. Além disso, de acordo com Ballard [Ballard 2007], o desenvolvimento de aplicações para dispositivos móveis precisa considerar três fatores fundamentais: o contexto móvel, o usuário móvel e a aplicação móvel que são discutidos nas próximas subseções.

2.2.1 Contexto Móvel

O fator contexto móvel está associado a limitações dos dispositivos, tais como pouca bateria, diferentes tipos de conectividade sem fio (e.g., Wi-Fi – Wireless Fidelity [IEEE Computer Society 2007], GPRS – General Packet Radio Service [Cai e Goodman, 1997] e 3G [Collins. e Smith 2002] [Andrade et al. 2006]), teclados limitados, telas pequenas, resultando em limitações de velocidade, processamento e memória bem mais do que nas aplicações tradicionais.

Dentre as características do contexto móvel, uma importante a ser considerada é que o usuário tipicamente carrega o dispositivo o tempo todo. Isso faz com que o dispositivo seja um dos principais fatores para entender a complexidade do contexto móvel.

Um dispositivo móvel pode ser definido como um computador portátil que o usuário pode carregar e que se comunica através de uma tecnologia sem fio. A facilidade de transportar (também chamada de portabilidade) e o tipo de conectividade justificam a “mobilidade” do dispositivo. São exemplos de dispositivos móveis: celulares, *smartphones*, PDAs, *tablet PC* (Personal Computer), *laptops*, dentre outros.

De acordo com Lee [Lee et al. 2005], dois fatores que afetam a facilidade de um usuário móvel transportar um dispositivo móvel são o tamanho e o peso do dispositivo. Os dispositivos maiores, tais como *laptops* são difíceis de transportar e por esse motivo, muitas vezes, são usados como computadores de mesa. Já os dispositivos menores como, por exemplo, os celulares são mais difíceis de serem utilizados. Além disso, quanto maior o peso do dispositivo mais difícil de transportá-lo.

Em virtude disso, na literatura são encontrados esforços na tentativa de melhor classificar os dispositivos móveis de acordo com as suas características (e.g., tamanho, peso, funcionalidade e custo). As classificações são apresentadas a seguir.

Segundo Weiss [Weiss 2002], os dispositivos são difíceis de quantificar e parecem atuar mais como um *laptop*. Sua classificação, que é chamada de “a continuação do computador pessoal”, é descrita em quatro categorias principais que são apresentadas na Figura 2.1.

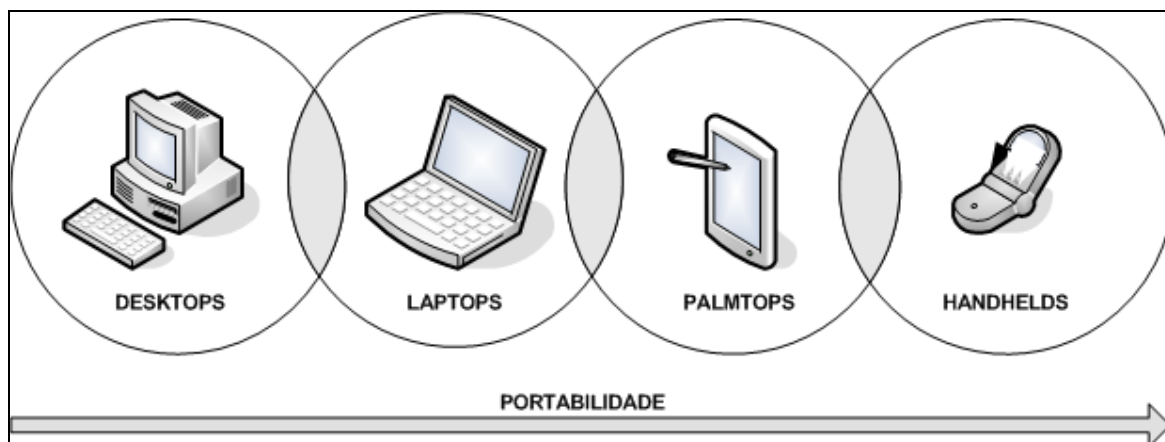


Figura 2.1 – A continuação do computador pessoal. Adaptada de [Weiss 2002]

Conforme a ilustração da Figura 2.1, o tamanho do computador pessoal diminui para a direita à medida que sua portabilidade aumenta. Os *laptops* são os primeiros computadores aptos a ter mobilidade, porém ainda são pesados o suficiente e, por isso, são utilizados sobre a mesa. Já os *palmtops* se parecem um pouco com os *laptops*, mas são significativamente menores e cabem dentro de um bolso ou uma bolsa. Os *handhelds* (computadores de mão) são bem pequenos e funcionam melhor quando segurados por uma das mãos. Portanto, os *handhelds* são extremamente portáteis e, segundo Weiss [Weiss 2002], devem atender as seguintes características:

- Operar sem cabos, exceto temporariamente, para carregar a bateria ou transferir dados para um *desktop*.
- Serem facilmente usados com uma única mão, sem a necessidade de estar apoiado sobre a mesa, e
- Permitir a adição de novas aplicações e/ou suportar conexão com a internet (e.g., WAP – Wireless Application Protocol [Barnes 2003]).

No entanto, os dispositivos também podem ser desenvolvidos baseados nas necessidades do mercado. Dessa forma, eles também podem ser classificados para suprir essa demanda. De acordo com Ballard [Ballard 2007], no que se refere à “Taxonomia do Dispositivo”, eles podem ser classificados de:

- **Trabalho de propósito geral:** dispositivos de múltiplos propósitos, como aqueles que são utilizados apenas durante o trabalho.

Esses dispositivos são basicamente os *laptops* ou computadores *tablet* com sistemas

operacionais baseados em sistemas operacionais completos para *desktop*.

- **Entretenimento de propósito geral:** dispositivos de múltiplos propósitos com foco no entretenimento, como aqueles que são utilizados quando o entretenimento é aceitável.

Esses dispositivos têm um conjunto de características de entretenimento baseadas em segmentos de mercado. Dependendo do foco do mercado, eles podem ter, por exemplo, como função principal os jogos e como característica secundária a mídia (e.g., vídeo, música). O dispositivo também pode ser baseado em “palavra escrita” (do inglês “written word”) permitindo ao usuário trabalhar com enigmas, ler livros digitais e navegar pela internet.

Além da diferença na interface do usuário, o dispositivo também pode apresentar diferenças no *design*. Por exemplo, dispositivos focados em comunicação de voz têm um alto-falante, um teclado numérico e um microfone. Já os dispositivos focados em jogos têm controles como periféricos de entrada.

Dessa forma, embora os dispositivos de entretenimento possam ter características secundárias, sua interface e seu *design* são de acordo com sua função principal.

- **Comunicações e controle de propósito geral:** dispositivos de múltiplos propósitos pessoal são aqueles usados para se comunicar através de voz e texto, e também para controlar aplicações de automação de casa e finanças.

Um dispositivo de comunicação é um dispositivo cujo objetivo primário é a comunicação. Ele é representado pelo Dispositivo Pessoal de Comunicações (PCD - Personal Communications Device), e se distingue dos computadores do tipo *desktop* nas quatro características, a seguir: (i) **pessoal**, o dispositivo pertence apenas a uma pessoa que carrega o dispositivo praticamente o tempo todo; (ii) **comunicativo**, o dispositivo envia e recebe mensagens (e.g., SMS – Short Message Service e MMS – Multimedia Messaging Service); (iii) **handheld**, pode ser segurado com uma das mãos e prontamente colocado no bolso; e (iv) **disponível**, o dispositivo está sempre apto a se comunicar com rede mesmo que ele esteja no estado ocioso ou de espera. A Figura 2.2 apresenta um resumo das características do PCD e como elas se relacionam.

- **Alvo:** dispositivos direcionados para a realização de uma ou um número muito pequeno de tarefas que variam de acordo com seu propósito.

As funções desses dispositivos são frequentemente incluídas em outros dispositivos.

São exemplos de dispositivos alvo: câmeras digitais, relógios, televisões, rádios, tocadores de música, máquinas de cartões de crédito, máquinas de auto-atendimento e escâner de código de barras.

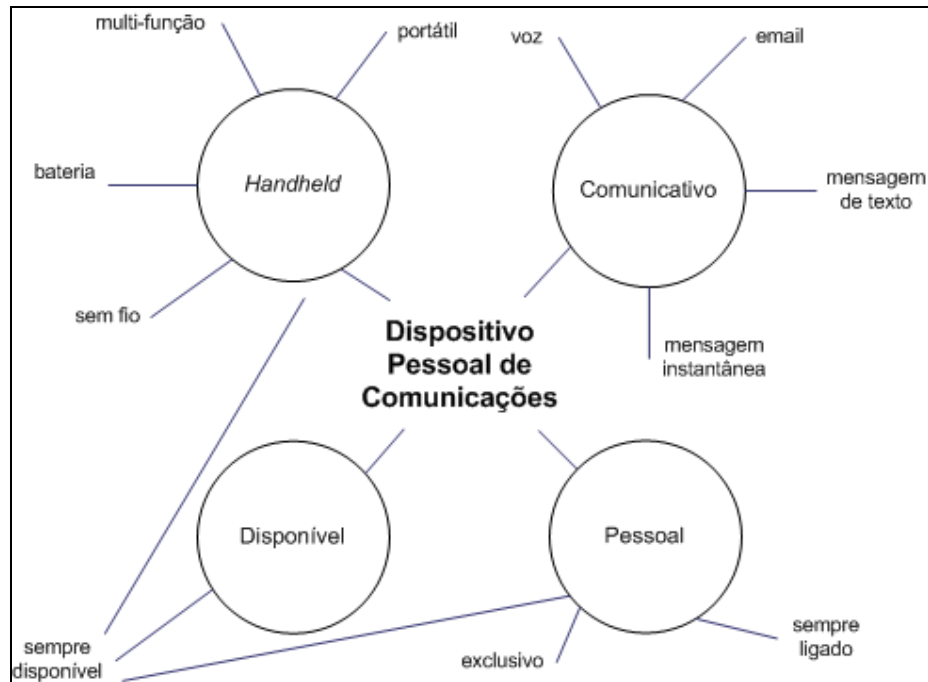


Figura 2.2 – Características PCD. Adaptada de [Ballard 2007]

Os dispositivos do tipo propósito geral (e.g., trabalho, entretenimento e comunicação) têm uma tendência a agrupar formas previsíveis do computador, tais como entrada de texto, cursor e uma tela. Essa combinação varia dependendo do tipo de dispositivo. Em contrapartida, os dispositivos alvo têm menos restrições quanto à forma e podem ser projetados perfeitamente para as tarefas que eles suportam).

A variedade de dispositivos móveis no mercado nos leva a diferentes classificações, como as que foram apresentadas acima. Com o intuito de simplificar as classificações apresentadas, a Figura 2.3 ilustra uma classificação genérica dos dispositivos móveis em relação à forma e a comunicação. Conforme essa figura, os dispositivos móveis são classificados em: (i) dispositivos maiores que podem ser apoiados sobre a mesa, possuem tela maior e teclado completo (e.g., *laptops* e *netbooks*) e que não possuem a funcionalidade telefone (sem comunicação de voz); (ii) dispositivos menores do tipo *handheld* (e.g., Tablet PC, Palms, PDAs, *smartphones* e celulares) que podem ser segurados com uma das mãos e operados pela outra¹. Os *handhelds* podem ou não

¹ Note que alguns modelos de *Tablets* não são considerados *handhelds*, pois são operados com o apoio dos braços em vez das mãos, além de possuírem uma tela maior quando comparados com os outros *handhelds*.

possuir comunicação de voz. Como muitas das limitações impostas pelos dispositivos menores (e.g., memória, processamento, armazenamento, tela, bateria) não se aplicam aos dispositivos maiores, considera-se nesta dissertação os dispositivos móveis pequenos do tipo *handheld* com comunicação de voz (e.g., PDAs, *smartphones* e celulares).

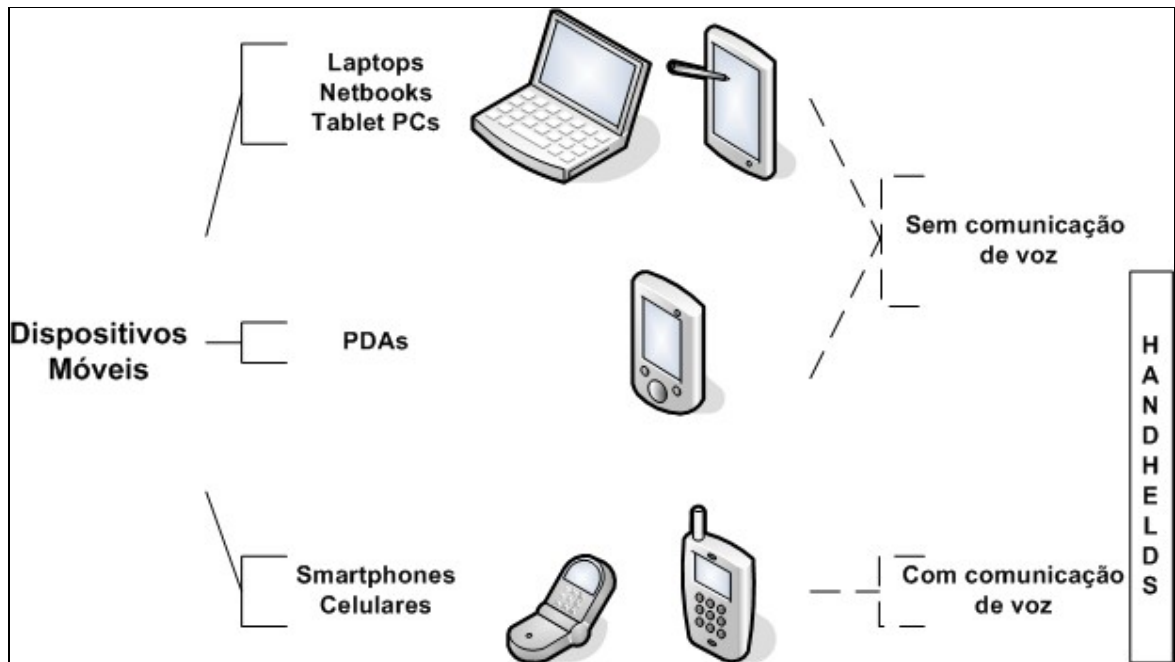


Figura 2.3 – Classificação dos dispositivos móveis em relação ao tamanho da tela

2.2.1.1 Dispositivos Móveis e Desktop

Apesar de os dispositivos móveis serem práticos, fáceis de transportar e, dessa forma, permitirem a mobilidade para o usuário, eles ainda possuem muitas limitações quando comparados a um computador de mesa. Limitações, tais como pouca *memória*, baixa capacidade de *processamento*, pouca duração da *bateria*, pouca capacidade de armazenamento e conexão sem fio mais lenta, um teclado pequeno para a entrada de informações que são visualizadas em uma *tela* também pequena acarretam em problemas inerentes ao ambiente móvel. Essas limitações são descritas a seguir:

- **Memória:** A capacidade de memória dos dispositivos móveis é bem menor que dos computadores de mesa (também chamados de computadores desktop). Aplicações que requerem uma quantidade maior de memória RAM para serem executadas nos dispositivos móveis como, por exemplo, os jogos que utilizam recursos gráficos.

- **Processamento:** a velocidade do processamento dos dispositivos móveis é bem menor do que os computadores de mesa. Os PDAs podem ter velocidade mais rápida, mas não se comparam a velocidade multigigahertz dos processadores dos computadores de mesa [Leggett et al. 2007].
- **Tela:** as restrições físicas dos dispositivos móveis, especialmente em relação à tela, tais como o pequeno tamanho e as baixas resoluções. O pequeno tamanho de tela pode fazer com que a maioria das páginas WWW (World Wide Web) quando visualizadas nos dispositivos possam ser não-navegáveis, não-amigáveis ou no pior caso completamente ilegíveis. As baixas resoluções (e.g., 128 * 128 pixels) podem degradar a qualidade da informação exibida na tela do dispositivo, além de que as diferentes resoluções dos dispositivos podem causar diferentes resultados para o teste de usabilidade. Ambas as situações podem consideravelmente afetar a usabilidade da aplicação móvel [Zhang and Adipat 2005].
- **Bateria:** os dispositivos móveis são alimentados por bateria e o seu tempo de vida é limitado. De acordo com Ballard [Ballard 2007], dependendo da tecnologia que a aplicação é desenvolvida, o consumo de bateria pode variar. Por exemplo, as mensagens de texto, para um número limitado de interações, usam muito pouca bateria. Em contraste, as mensagens multimídia utilizam mais bateria tanto devido ao tempo maior de download quanto porque o usuário vai passar mais tempo visualizando uma foto do que lendo uma simples mensagem de texto.
- **Capacidade de armazenamento:** a capacidade de armazenamento varia de dispositivo para dispositivo. A possibilidade de expansão da memória interna através dos cartões de memória proporcionou um aumento na quantidade de recursos que podem ser usados na aplicação. No entanto, o armazenamento ainda é um fator limitante porque segundo Ballard [Ballard 2007], o usuário móvel precisa de armazenamento persistente quer localmente ou em um servidor que varia de acordo com a natureza dos dados e os requisitos da aplicação.
- **Largura de banda:** A força do sinal e a velocidade de transferência de dados na rede sem fio podem variar dependendo da localização do usuário móvel. Como lidar com as várias condições da rede sem fio deve ser considerado no estudo de usabilidade da aplicação móvel [Zhang and Adipat 2005].

Além das limitações apresentadas acima, cada tecnologia tem seus prós e contras que podem influenciar nas limitações apresentadas acima. A plataforma adotada para o ambiente de

execução e desenvolvimento das aplicações móveis varia de sistemas operacionais: Palm, Windows Mobile, Symbian a linguagens de programação: Java Micro Edition – JME, Symbian Native C++, Brew – Binary Runtime Environment for Wireless, Python para Symbian e Flash Lite.

Outro fato é que algumas limitações já estão deixando de ser um desafio. Por exemplo, dispositivos já possuem uma boa capacidade de armazenamento que ainda pode ser expandida pelos cartões de memória (cartões de até 32MB) e bateria de fabricantes como a Nokia que tem uma boa duração. Porém, embora algumas limitações sejam minimizadas, a grande maioria dos dispositivos vendidos no mercado ainda apresenta essas limitações.

Além disso, as finalidades de uso do dispositivo móvel e do computador de mesa se divergem. O usuário móvel deseja informação rápida diferente do usuário *desktop* que desempenha tarefas mais complexas e, dessa forma, uma pesquisa de forma mais demorada pode levar horas sem prejuízo de custo e tempo.

De acordo com Ballard [Ballard 2007], a diferença fundamental do *design* dos dispositivos móveis para o *design* de outras plataformas é o fato de o usuário carregar o dispositivo o tempo todo e isso tem muitas implicações, dentre as quais o autor cita:

- O fato dos dispositivos serem pequenos e possuírem pouca bateria, diferentes tipos de conectividade sem fio, além de teclado e telas pequenos.
- Qualquer informação ou entretenimento pode ser desejável.
- Capacidades, tais como conexão sem fio, tamanho pequeno e restrições de energia, têm feito os dispositivos terem velocidades bem menores, processadores menores e bem menos memória do que os computadores de mesa.
- A tela de um dispositivo móvel permite apenas uma janela por vez, o que dificulta compartilhar informações entre as aplicações do dispositivo.
- Um dispositivo pessoal tem a necessidade de se suprir a necessidade do usuário.

Em [Kaikkonen e Roto 2003] também é apresentado algumas características específicas dos dispositivos móveis, dentre as quais se destacam:

- Grande variação nas dimensões da tela;

- O crescimento do número de cores da tela;
- A entrada de texto é bem menor do que um teclado completo de PC;
- Usualmente não existe *mouse* para ativar um objeto;
- Alguns dispositivos somente suportam barra de rolagem vertical;
- As teclas do tipo *softkeys* são comumente usadas para ativar comandos, mas o número e o propósito das mesmas variam entre os dispositivos de diferentes fabricantes;
- A transferência de dados entre o dispositivo e o servidor é baixa;
- O contexto de uso é bem mais difícil de premeditar do que uma aplicação *desktop*; e
- O usuário pode ter que pagar pelos trafegados na rede.

Nesse contexto, percebe-se que existem muitas diferenças entre um dispositivo móvel e um computador *desktop*. Weiss [Weiss 2002] faz uma comparação dos dispositivos móveis e computador de mesa conforme pode ser observado na Tabela 2.1.

Tabela 2.1 – Comparação do dispositivo móvel e computador de mesa. Adaptada de [Weiss 2002]

	Dispositivo Móvel²	Computador de Mesa
Tela	Em média, até 4.13'' e no máximo 800 x 480 pixels	Em média, a partir de 14'' e no mínimo 640 x 800 pixels
Capacidade de Armazenamento	Normalmente pouca. Embora, atualmente, alguns dispositivos possuem de memória para expansão de 32GB	Praticamente ilimitada. (e.g., DVD, Disco Rígido e <i>pendrive</i>)
Memória	Em média, no máximo, 128MB de RAM Bem pequena e limitada.	Em média, no mínimo, de 128 MB de RAM Bem superior aos dispositivos móveis
Processamento	Mais lento	Rápido
Energia	Boa duração, mas é necessário recarregar a bateria após um período de tempo que pode variar de acordo com a tecnologia de acesso de rádio usada, configuração e uso	O usuário precisa estar conectado a uma fonte de energia através de um cabo de alimentação
Conectividade	Sem fio (e.g., Wi-Fi, GPRS e 3G) Mais lenta e instável	Com fio e sem fio (e.g., Wi-Fi e 3G) Rápida e estável
Mobilidade	Pode ser usado facilmente com o usuário em movimento Passível de interrupção	Precisa está apoiado sobre a mesa
Dispositivos de Entrada	Teclado pequeno e limitado, caneta, tela sensível ao toque, botões para navegação, microfone, câmera e cartão de memória	Teclado completo, mouse, tela sensível ao toque, <i>webcam</i> e microfone

2.2.2 Usuário Móvel

O usuário *desktop* interage somente com o computador enquanto o usuário móvel tem um grau de interação maior, uma vez que possui mobilidade, podendo interagir com o ambiente como um todo. É importante ressaltar que o usuário em ambos os casos é uma única pessoa, dessa forma, não existe uma diferença particular entre esses dois usuários. O fato de que os dispositivos de uma maneira geral têm custos menores do que um computador nos leva a dizer que hoje um usuário *desktop* é um subconjunto de um usuário móvel.

A característica chave do usuário móvel é a **mobilidade**, pois ele pode se mover enquanto usa uma aplicação. A conectividade que o contexto móvel proporciona ao usuário é de

² As informações dos dispositivos móveis são baseadas em dados levantados a partir de uma pesquisa feita em um dos maiores fabricantes de dispositivos móveis do tipo *Handheld*

que ele sempre está **disponível** para família, amigos, clientes, dentre outros. Nesse cenário, o usuário móvel está sempre presente e disponível através do telefone do dispositivo, assim ele também está **sociável** para as pessoas que estão ao seu redor. Geralmente, o dispositivo é **pessoal** e pertence a apenas a uma pessoa e dessa forma o usuário é **identificável** através do dispositivo. O ambiente afeta em como o dispositivo é usado, o ideal seria que o dispositivo soubesse o que o usuário móvel está em uma reunião de negócios, em uma viagem ou em outra atividade, assim podemos dizer que o usuário móvel é **contextual** [Ballard 2007].

Outra característica do usuário móvel é que ele é **passível de interrupção**. Por exemplo, o usuário ao usar o dispositivo pode fazer apenas uma coisa por vez, se o dispositivo é interrompido por um recebimento de uma chamada e o usuário está digitando uma mensagem naquele momento, ele não terá a chance de terminar e salvar a mensagem quando é interrompido. Outro exemplo em um contexto social, é que o ônibus não pode esperar o usuário terminar uma tarefa ou uma conversa. Dessa forma, o usuário pode ser interrompido quando usa o dispositivo.

Todas essas características proporcionadas pelos dispositivos nem sempre agradam a todos os usuários móveis. Por exemplo, ser interrompido em um dispositivo pequeno com uma tela pequena e que mostra apenas uma janela por vez pode deixar o usuário móvel irritado. Alguns deles se sentem desconfortáveis com tanta disponibilidade e preferem desligar o dispositivo móvel ou deixá-lo no modo silencioso durante uma reunião de trabalho ou até mesmo em um momento de lazer. Atender chamadas telefônicas durante uma conversa pessoal tem sido muito comum entre os usuários móveis, embora alguns ainda considerem esse ato muito deselegante.

Dessa forma, uma aplicação móvel deve considerar essas características e tratá-las de forma adequada para que o usuário não se sinta prejudicado. No caso das interrupções, a aplicação deveria permitir que o usuário voltasse para o estado da aplicação anterior as mesmas.

2.2.3 Aplicação Móvel

A aplicação móvel é uma aplicação que executa em um dispositivo móvel. Ela é utilizada em vários domínios de aplicação por um usuário (que pode estar em movimento).

A maioria das aplicações móveis é criada como uma versão miniatura de aplicações similares *desktop*. Contudo, segundo Ballard [Ballard 2007], para que uma aplicação móvel seja bem sucedida ela tem que considerar não só a portabilidade do ambiente fixo para o móvel, mas também as características que se sobrepõem parcialmente em relação às aplicações *desktop* (também conhecidas como aplicações para PC) e complementá-las, em vez de ser apenas ser um

subconjunto dessas aplicações.

Dessa forma, as aplicações móveis possuem as limitações das aplicações *desktop*, todas as limitações do ambiente móvel e ainda as limitações dos desenvolvedores e projetistas que fazem a mudança do contexto fixo para o móvel.

Quando se vai testar uma aplicação para dispositivo móvel, o testador geralmente se pergunta o que deve ser testado na mesma e qual é a diferença do teste de uma aplicação móvel para uma aplicação *desktop* (também chamada de aplicação para PC).

Segundo Ballard [Ballard 2007], o ambiente *desktop* compreende um conjunto de características, tais como, todos possuem uma tela colorida muito grande de no mínimo 800x600 pixels, um teclado completo, um mouse, caixas de som, aplicações residindo em janelas e conectividade geralmente rápida (também pode ser baixa em torno de 30 Kbs). Além disso, o usuário de uma aplicação *desktop* está sentado a uma mesa ou ao menos com um computador no colo. Existe uma superfície de trabalho, e as mãos e a atenção está voltada para o computador. A interação com outras pessoas ocorre apenas através do computador, geralmente não existem pessoas ao redor do computador.

Já no ambiente móvel as regras são diferentes, não por causa da falta de normas, mas devido às grandes variações das necessidades dos usuários móveis. As diferentes capacidades dos celulares, dos *smartphones* e de outros tipos de dispositivos móveis levam a um ambiente variável. Dessa forma, de acordo com Ballard [Ballard 2007], uma aplicação móvel consiste de:

- Um dispositivo, um navegador, um ambiente de aplicação e suas capacidades.
- Um usuário, usando qualquer conjunto de dispositivos, que poderia estar andando de trem, sentado em uma reunião, sentado em um restaurante, caminhando na rua ou focado em outras tarefas.
- Uma ou mais plataformas de aplicação, que pode incluir navegadores web, ambiente de aplicação (e.g., Brew, Palm, Windows Mobile, Symbian, JME), mensagens (e.g., email, SMS, MMS e mensagens instantâneas), ambientes de Mídia (tipos de música e visualizadores de vídeo) e novas capacidades que podem vir a ser disponibilizadas.
- Uma ou mais interfaces de saída do dispositivo incluindo tela, caixa de som, IrDA, Bluetooth, Wi-Fi, dentre outros.

- Uma ou mais interfaces de entrada do dispositivo, tais como teclado limitado, tela sensível ao toque, microfone, câmera, leitor RFID (Radio-Frequency IDentification), infravermelho, bluetooth, Wi-Fi.
- Opcionalmente um servidor que complementa a aplicação móvel e adiciona informação ou funcionalidade.
- Interfaces entre servidor de aplicação e outras fontes de informação.
- Uma rede um operador sem fio que habilita algumas das tecnologias citadas, conecta o usuário a internet, vende outras aplicações e outros serviços e frequentemente define o que pode ser acoplado ou não a rede.

2.2.3.1 Tipos de Aplicações Móveis

O tipo de aplicação móvel também pode influenciar na usabilidade. A seguir serão listados os principais tipos de aplicações móveis apresentados em [Pocatilu 2008] [Weiss 2002] e problemas de usabilidade relacionados com as suas interfaces.

Web Móvel: refere-se ao navegador baseado em serviços *web*, como a WWW, WAP e i-Mode (Japão), utilizando um dispositivo móvel, como um telefone celular, PDA ou outro dispositivo portátil ligado a uma rede pública.

A maioria das páginas *web* quando visualizadas nos dispositivos móveis não são bem renderizadas, especialmente pela grande quantidade de informação que possuem e, com isso, o texto pode ser visualizado com cortes ou as imagens distorcidas. Dessa forma, o acesso a *web* móvel ainda sofre com problemas de usabilidade.

Comércio eletrônico (e-commerce): O comércio eletrônico para a *web* sem fio, muitas vezes referenciado como *m-commerce*, é, atualmente, modelado depois de comércio eletrônico da web para *desktop*. No entanto, pouco gasto é feito pelos usuários via dispositivos móveis, porque os problemas de usabilidade, tais como (i) entrar com informações nos dispositivos móveis é entediante, (ii) é impossível representar uma mercadoria de uma maneira eficiente em uma tela pequena, (iii) as pessoas estão em movimento e improvável elas naveguem para compras a partir de dispositivo móvel.

Jogos: Um jogo móvel é um vídeo *game* que pode ser jogado em computadores portáteis, tais como celular, *smartphone* e PDA. Ele fornece um mecanismo fácil e rápido de entretenimento quando os usuários estão entediados. De acordo com Ballard [Ballard 2007], o *design* dos jogos é

diferente do design das outras aplicações que focam em informações. As aplicações que não são jogos focam em eficiência, já os jogos focam em entretenimento. Além disso, enquanto os jogos variam muito, a estrutura de navegação para suportar o jogo não. Questões, tais como o *design* do jogo (e.g., tela de introdução, menu principal, opções de tela e as tela de jogos – gráficos, ações dos jogos, menu do jogo e tecla de entrada), som, vibração do som e jogos com multijogadores fazem qualquer jogo completamente funcional.

Aplicações de negócios: as aplicações de negócios podem se referir a uma ampla audiência de negócio (e.g., checar o status de um vôo, banco online e aplicações de serviço ao cliente) ou aplicações desenvolvidas para automatizar uma necessidade específica de negócios (e.g., preparar um monitoramento de um paciente na área de saúde). Ambas requererem alta usabilidade e suporte técnico, tais como acomodar o início e a parada de um processo sem perder dados, garantir a consistência nas interfaces de usuário *desktop web* e manter a comunicação dos sistemas de modo que os perfis disponíveis para *desktop* também sejam para a *web* sem fio.

Aplicações para produtividade: incluem os calendários, calculadoras, agendas, bloco de notas, planilhas e serviços de diretório.

Aplicações para comunicação: incluem *email*, vídeo, telefone, bate-papo interativo e sistemas de fóruns de discussão e mensagens.

Utilitários: incluem gerenciadores de perfis, tarefas, de chamada e arquivos.

2.3 Conclusão

Este capítulo apresentou os principais conceitos acerca do Ambiente Móvel incluindo as suas principais características: o contexto móvel, a aplicação móvel e o usuário móvel. O principal objetivo deste capítulo foi fornecer o embasamento teórico sobre o Ambiente Móvel para um melhor entendimento da complexidade dos testes em aplicações móveis. Além disso, os tipos de aplicações móveis foram apresentados e os problemas de usabilidade relacionados a essas aplicações também foram discutidos.

No Capítulo 3 são apresentados os principais conceitos relacionados com a área, além de mostrar em quais aspectos do ambiente móvel os testes das aplicações móveis se diferencia do teste de aplicações *desktop*.

Capítulo 3

Testes de Software

Neste capítulo são apresentados os conceitos de teste de software mais relevantes para esta dissertação. Na Seção 3.1 é apresentada uma visão geral sobre testes de software, sendo que na Seção 3.2 são apresentados os conceitos de teste de software. Na subseção 3.2.1 são definidos os casos de testes e na subseção 3.2.2, os tipos de testes que são utilizados nesta dissertação. A Seção 3.3 apresenta testes de software para aplicações móveis e as principais diferenças do mesmo para os testes de software tradicionais. A Seção 3.4 foca em usabilidade e experiência do usuário e como elas se relacionam. A Seção 3.5 aborda o teste de usabilidade para as aplicações móveis e quais aspectos do ambiente móvel podem ser avaliados por ele e, finalmente, a Seção 3.6 apresenta as principais considerações deste capítulo.

3.1 Verificação e Validação

O teste é uma técnica dinâmica de verificação e validação (V&V) que envolve executar um programa com um conjunto de entrada de dados e verificar se está de acordo com o esperado.

Os termos verificação e validação não possuem o mesmo significado e, de acordo com Sommerville [Sommerville 2007], verificação verifica se o software está de acordo com suas especificações e validação checa se o software atende às expectativas do cliente. V&V abrange uma série de atividades de garantia da qualidade do software (do inglês Software Quality Assurance - SQA) e também pode ser referenciada como Validação, Verificação e Testes (V&VT).

As atividades de V&VT devem ser conduzidas durante todo o processo de desenvolvimento de software, desde a sua concepção e podem ser divididas como estáticas e dinâmicas [Delamaro et al. 2007]. Nas técnicas estáticas, tais como as inspeções de software não é necessário a execução de um programa ou modelo executável. As inspeções verificam qualquer representação legível do sistema, tais como os documentos de requisitos e os diagramas de projeto, mas são mais utilizadas para inspeção de código-fonte. Já as dinâmicas, como, por exemplo, os testes de software devem ser conduzidas pela execução de um programa.

As inspeções são bastante usadas no processo de software, mas elas não podem verificar como o software realmente funcionará, questões como confiabilidade e desempenho só poderão ser tratadas com os testes de software [Sommerville 2007].

O foco deste trabalho são testes para aplicações móveis, e nesse contexto é essencial testar a aplicação rodando tanto em emuladores como no próprio dispositivo, sendo assim, as inspeções de software por ser um processo estático não são adequadas e, portanto, não são consideradas nesta dissertação.

3.2 Conceitos de Testes de Software

Existem diferentes definições para testes de software na literatura, dentre as quais são:

- O processo de executar um programa com o objetivo de encontrar erros [Myers 2004];
- O processo de avaliar um sistema ou um componente de um sistema por meios manuais ou automáticos para verificar se ele satisfaz os requisitos especificados ou identificar diferenças entre resultados esperados e obtidos [Koscianski e Soares 2007];
- Uma atividade desempenhada para avaliar a qualidade do produto e melhorá-lo uma vez que identifica defeitos e problemas [IEEE 2004];
- Um processo concorrente no ciclo de vida da engenharia de software a fim de medir e melhorar a qualidade da aplicação que está sendo testada [Craig and Jaskiel 2002].

As definições apresentadas acima enfatizam duas características: encontrar erros e qualidade. A primeira característica é relacionada aos diferentes significados que podem causar confusão dos seguintes termos: defeito, falha e erro.

Defeito é uma imperfeição do produto, ele se refere a algo que está implementado no código de maneira incorreta. *Falha* é o resultado errado provocado por um defeito ou condição inesperada. Por esse motivo, um defeito pode ocasionar a ocorrência de um *erro* durante a execução de um programa e levar a um estado inconsistente ou inesperado e o mesmo pode levar a uma falha fazendo com que o resultado produzido seja diferente do resultado esperado [Delamaro et al. 2007]. Esses termos não são consenso na literatura e muitas vezes são empregados de formas diferentes. Nesta dissertação é utilizada a palavra “erro” que pode significar qualquer um dos termos definidos.

O segundo fator é que se a atividade de teste for planejada de forma sistemática e rigorosa, ela pode ser utilizada como um dos parâmetros para estimar a confiabilidade e qualidade do software construído [Koscianski e Soares 2007]. Por esse motivo, o teste de software também pode ser considerado como um dos componentes da Garantia de Qualidade de Software (do inglês Software Quality Assurance - SQA). Segundo Molinari [Molinari 2005], o teste de software garante a qualidade ao diminuir a quantidade de erros no produto final. No entanto, [Kruchten 2003] afirma que o papel do teste não é assegurar a qualidade, mas sim avaliá-la. Neste trabalho, é considerado que os testes de software melhoram a qualidade do software.

Os testes podem somente mostrar a presença de erros, não a sua ausência [Dijkstra et al. 1972]. Dessa forma, nem todos os defeitos são encontrados durante os testes, e mesmo que não sejam descobertos, não quer dizer que eles não existam, eles estarão na aplicação. Portanto, pode-se dizer que um dos principais objetivos dos testes de software é minimizar os riscos de aparecer um novo defeito e não de eliminá-los totalmente porque não é possível testar todas as situações. Minimizar a ocorrência de problemas ao máximo é o ponto de equilíbrio do teste e um parâmetro para o testador saber até quando testar o programa.

Contudo, a meta do testador é encontrar erros o mais cedo possível. Segundo [Molinari 2005], o custo de um defeito descoberto no início custa cerca de 10 centavos de dólar e quando é encontrado mais perto do release custa em torno de 100 dólares. Logo, quanto mais cedo identificar o erro menor será custo para consertá-lo.

Na literatura existem relatos de grandes erros de software, pode-se citar o caso do Mars Polar Lander da NASA em 1999, que foi totalmente perdido na tentativa de pousar em Marte. A missão foi um fracasso e teve um prejuízo de U\$ 165 milhões. A investigação concluiu que um bit inesperado no conjunto de instruções do programa foi o motivo do fracasso. Outro caso foi o do Ariane 5, em julho de 1996, que explodiu 4 segundos após do seu lançamento, depois de uma década de desenvolvimento e U\$ 7 bilhões de investimentos [Almeida et al. 2007]. A causa da queda foi um *overflow* na conversão de tipos. Percebe-se que essas aplicações falharam devido ao processo de testes não ser tão rigoroso.

Um relato ocorrido com dispositivos móveis, especificamente em aparelhos celulares, foi que devido a um erro, um determinado telefone celular só realizava ligações de emergência depois de checar a posição do aparelho através do Global Positioning System (GPS). Na ausência do sinal, o celular bloqueava as ligações, por exemplo, para 190. A fabricante teve que corrigir a falha em trinta mil aparelhos [Barreto 2009].

Mesmo sabendo das conseqüências apresentadas, os testes são muitas vezes colocados em segundo plano e não são realizados de forma planejada devido ao seu alto custo. A atividade de testes pode custar cerca de 50% dos custos dependendo do tipo de sistema [Pressman 2006]. Portanto, além do processo de teste ser baseado na intuição e experiência dos engenheiros de software, ele também tem que ser planejado.

Segundo [Molinari 2005], o planejamento de testes facilita a organização, objetividade, previsibilidade e gerenciamento do processo de testes, mas tem como obstáculos o prazo, que limita o escopo. Outro fator constatado é que quanto mais cedo a equipe de teste é envolvida, mais problemas poderão ser encontrados antes de serem transformados em código. A equipe não precisa esperar que o sistema seja construído para começar a testar, pois os defeitos de uma aplicação podem ser descobertos a partir do momento que se planeja até o fim do ciclo de vida da aplicação.

Por esse motivo, o teste de software deve ser realizado em fases diferentes do ciclo de desenvolvimento software, pois tudo que está sendo desenvolvido segue um processo de teste, dependendo de qual fase o software se encontre. O teste não é uma fase única e deve ser tratado como um projeto dentro de outro projeto [Molinari 2005]. Além disso, de acordo com Graham [Graham 2002], ter profissionais de teste envolvidos na análise dos requisitos é uma das melhores maneiras de assegurar bons requisitos, pois os requisitos terão algum critério de qualidade consistente.

3.2.1 Casos de Teste

De acordo com [Inthurn 2001], um caso de teste é um documento que descreve uma entrada, ação ou evento e um resultado esperado, a fim de determinar se uma funcionalidade da aplicação está executando corretamente ou não. O objetivo do processo de projeto de casos de testes é criar um conjunto de casos de testes que são eficazes o suficiente para descobrir os erros de um programa.

Os casos de teste podem ter as seguintes abordagens: teste baseado em requisitos, teste de partições e teste estrutural [Sommerville 2007]. Teste baseado em requisitos é quando os casos de testes são elaborados para testar os requisitos do sistema. Teste de partições é feito com grupos de dados que possuem características comuns, as partições que são de entrada e de saída. A abordagem Teste estrutural é quando os casos de teste são feitos a partir do conhecimento da estrutura interna do sistema de modo que eles exercitem todas as partes deste.

3.2.2 Tipos de Teste

As duas estratégias gerais de teste são o teste de caixa-preta (do inglês Black-Box Testing) e o teste de caixa-branca (do inglês White-Box Testing) que segundo Molinari [Molinari 2005], estão na dimensão de “Como vou testar”. São também classificadas na literatura como Critérios de Teste [Santos-Neto 2006]. O teste de caixa-preta também conhecido como teste funcional é um método usado para testar um programa sem conhecer a sua estrutura e comportamento interno. O teste de caixa-branca também conhecido como teste estrutural é feito a partir do acesso a estrutura interna do programa.

No entanto, existem diversos tipos de testes também citados como técnicas, fases ou métodos para se testar o software. É difícil encontrar na literatura uma classificação homogênea sobre os tipos de testes. Em [IEEE 2004] é apresentada uma classificação de acordo com níveis e objetivos, pois o teste pode ser aplicado a diferentes tipos de objetivos em diferentes níveis de teste. Existem três grandes níveis de testes, os quais são:

- *Teste de unidade:* verifica as funções do software de forma isolada. Elas podem ser pedaços de código, classes, subprogramas ou um componente feito de unidades fortemente relacionadas. São exemplos de testes do tipo caixa branca onde a estrutura interna (código) não é levada em conta [Bastos et al. 2007].
- *Teste de integração:* verifica se as unidades testadas individualmente funcionam corretamente quando integradas.
- *Teste de sistema:* se preocupa com o teste de todo o sistema, verifica se o software está de acordo com os requisitos especificados. É, geralmente, considerado apropriado para comparar requisitos não funcionais, tais como segurança, velocidade, acurácia e confiabilidade.

Contudo, existem muitas maneiras de se testar o software, dependendo do tipo de aplicação que se deseja testar, e, portanto, os objetivos do teste podem variar. Para testar propriedades do sistema como, por exemplo, medir confiabilidade, avaliação de usabilidade, aceitação, corretude, funcionalidade e conformidade, pode-se utilizar diferentes objetivos de testes [IEEE 2004]. Eles são descritos a seguir:

- **Teste de desempenho:** verifica se o software satisfaz os requisitos de desempenho específicos como, por exemplo, capacidade e tempo de resposta.

- **Teste de estresse:** exercita o software com a carga máxima projetada e também ultrapassando essa carga para ver como o mesmo se comporta.
- **Teste de regressão:** é um reteste seletivo de um sistema ou componente para verificar se as modificações não causaram efeitos indesejados.
- **Teste de usabilidade:** avalia como é fácil para os usuários finais a usar e aprender o software, incluindo documentação do usuário, suporte nas tarefas do usuário de forma eficaz e habilidade do sistema de se recuperar de erros dos usuários.
- **Outros objetivos de teste:** Teste de aceitação, teste alfa e beta, teste de conformidade, teste de regressão, teste de configuração, dentre outros.

Os objetivos de teste não devem ser aplicados isoladamente, eles devem ser usados de forma complementar para que o teste seja feito de forma mais completa e eficaz. Eles podem ser aplicados de acordo com a fase de desenvolvimento em que o software se encontra ou também podem ser utilizados de acordo com o tipo de aplicação a ser testada.

A seguir outros tipos de testes encontrados na literatura [Bastos et al. 2007] [Nguyen 2001] e que não são classificados em [IEEE 2004]:

- **Teste de Carga:** estuda como um programa lida com grande quantidade de dados, como por exemplo, cálculos e processamento excessivos [Nguyen 2001].
- **Teste de Interoperabilidade:** testa se um sistema ou componentes dentro de um sistema interagem e trabalham de forma transparente com outros sistemas ou outros componentes. De acordo com Bastos [Bastos et al. 2007], O teste de interoperabilidade avalia as condições de integração com outros sistemas e/ou ambientes.

Os níveis, objetivos, técnicas, fases e tipos de testes são referenciados nesta dissertação como tipos de testes.

3.3 Testes para aplicações móveis

O teste é uma fase obrigatória que tem como objetivo certificar que os dispositivos móveis são confiáveis, robustos e “livres de erros” [Mazlan 2006]. No entanto, na prática esta tarefa não é trivial e requer adaptações no processo de teste tradicional, tornando-o mais complexo. Essa complexidade envolve as questões do ambiente *desktop* e mais as limitações do ambiente móvel que devem ser incorporadas no processo de teste para aplicações móveis

[LioNBRIDGE 2006]. Por exemplo, durante o teste de aplicações móveis, os testadores enfrentam problemas típicos do ambiente móvel que são:

- Quais os tipos de testes necessários para verificar fatores do ambiente móvel (e.g., contexto móvel, usuário móvel e aplicação móvel)
- Quais são as características que devem ser testadas nas aplicações móveis em laboratório e em campo?
- Quais os fatores devem ser avaliados no teste de usabilidade da aplicação móvel?
- Como testar as aplicações móveis na diversidade de modelos de dispositivos móveis?

Em virtude disso, os testes das aplicações móveis devem ser planejados considerando os três fatores chave do ambiente móvel apresentados no Capítulo 2 (e.g., contexto móvel, usuário móvel e aplicação móvel). Primeiro, o contexto móvel que envolve as limitações dos dispositivos; segundo, o usuário móvel que está relacionado com a mobilidade e as outras características proporcionadas por esta; e por último, a aplicação móvel que executará em diferentes plataformas e poderá se comunicar com a rede.

De acordo com Ballard [Ballard 2007], as aplicações móveis, além de possuírem todas as limitações das aplicações *desktop*, incluem as limitações do ambiente móvel e ainda a limitação dos desenvolvedores e projetistas que precisam se adaptar para o contexto móvel. No entanto, Zeidler [Zeidler et al. 2007] afirma que os desenvolvedores ainda não têm muito conhecimento em desenvolver para dispositivos móveis e se concentram na implementação livre de erros em vez de focar em questões específicas do ambiente móvel.

Os testes nas aplicações móveis podem ser feitos através de emuladores que simulam a aplicação no próprio computador. Porém, segundo [Zeidler et al. 2007], o uso de emuladores não demonstrou ser um meio de substituição dos testes nos próprios dispositivos, portanto, a tarefa de testar no próprio dispositivo é um mandatório para a equipe de testes.

O processo de teste para as aplicações móveis é diferente das aplicações tradicionais pelas seguintes razões:

- **Aumento na quantidade de usuários móveis**

O número de pessoas que possuem dispositivo móvel tem crescido muito, assim como o acesso a internet a partir do mesmo. Já se pode dizer que o usuário *desktop* é um subconjunto do usuário móvel.

- **Menos experiência**

As pessoas geralmente têm menos experiência em usar o dispositivo para acessar a internet e outros aplicativos.

- **Tipos de dispositivos**

Na literatura, não existe um padrão para os diversos modelos de dispositivos móveis no que se refere ao uso, forma ou interfaces. As características variam de tamanho e resolução de tela, tipos de entrada do dispositivo (e.g., teclado ou tela sensível ao toque), diferentes formatos de teclado (e.g., numérico ou QWERT, teclas de navegação direcionais e *softkeys*) até outras funcionalidades (e.g., acelerômetro, câmera e mp3). Por causa disso, cada dispositivo pode apresentar a aplicação de uma forma diferente. Um dos desafios é garantir que a aplicação seja genérica o suficiente para se comportar de maneira semelhante em todos os modelos de dispositivos. .

- **Teste no emulador e dispositivo móvel**

As aplicações móveis podem ser testadas em dois ambientes: emulador no computador e dispositivo móvel. Dependendo do ambiente a aplicação poderá se comportar de maneira diferente.

No entanto, devido à diversidade de modelos de dispositivos móveis no mercado, é um desafio garantir que a aplicação seja genérica o suficiente para se comportar de maneira semelhante em todos os modelos de dispositivos.

- **Teste em laboratório e em campo:**

Os testes de aplicações móveis podem ser realizados em laboratório e em campo. O teste em laboratório é feito com um usuário sentado e um dispositivo móvel apoiado sobre uma mesa (ou um tripé) ou um emulador no computador executando a aplicação. O teste em campo pode ser feito com usuários reais [Nokia 2004] usando os dispositivos móveis em ambiente do mundo real, dessa forma, ele captura fatores específicos do ambiente móvel, por exemplo, interrupções, movimento físico, barulho, diferentes condições de iluminação só podem ser avaliadas em campo.

- **Interface ou domínio das aplicações móveis**

Existem diferentes categorias (e.g., *standalone*, baseada na *web* e distribuída) e tipos (e.g., jogos, web, corporativa, e-commerce, produtividade, comunicação e utilitários) para classificar as aplicações móveis conforme apresentado no Capítulo 2. Em virtude disso, pode-se dizer que os tipos e as categorias das aplicações móveis podem influenciar no

processo de teste, além disso, o tipo de teste (também referenciados por objetivos neste capítulo) das aplicações móveis também pode variar.

Em relação às categorias, segundo Pocatilu [Pocatilu 2008], o processo de teste para aplicações móveis do tipo *standalone* é similar para aplicações móveis desde que os testadores considerem a limitação dos recursos dos dispositivos móveis e poderiam ser inclusos os seguintes testes: teste funcional, teste de usabilidade, teste estrutural e teste de desempenho. Já para as aplicações distribuídas, os testes devem ser diferentes. O lado cliente e o lado servidor da aplicação necessitam ser testados a princípio de forma independente e então testados de forma integrada. No teste de aplicações móveis baseadas na web, a parte cliente requer testar as funcionalidades das aplicações, conteúdo, desempenho, carga, segurança e a base de dados, além da parte do servidor e do conteúdo que é enviado para o dispositivo móvel.

No entanto, o autor não leva em consideração o tipo das aplicações móveis. Por exemplo, testar um jogo móvel é diferente de se testar uma aplicação móvel corporativa independente da sua categoria. Embora não seja diferente todo o processo, mas fatores tal como jogabilidade do jogo na tela pequena do dispositivo móvel só é considerado no teste de jogo móvel. Além disso, o fato do processo de teste para aplicações móveis *standalone* ter que considerar as limitações dos dispositivos móveis, já o torna diferente do processo de teste para aplicações desktop. Outro ponto é que os tipos de testes sugeridos para as aplicações *standalone* e aplicações com interface web pelo autor, não são justificados e, portanto difícil de compreender o motivo para tal escolha.

- **Mobilidade do usuário**

Na maioria das vezes, os usuários móveis (definidos e caracterizados no Capítulo 2) usam a aplicação enquanto se movem e são passíveis de interrupções. O que resulta em uma plataforma de acesso muito menos previsível quando comparada as aplicações desktop. Além disso, a tarefa de navegar em uma aplicação móvel através de uma tela pequena com pessoas e objetos ao seu redor pode afetar na interação do usuário com a aplicação. A mobilidade não faz parte das aplicações desktop e dessa forma, não é tratada no seu processo de teste.

3.4 Usabilidade e experiência do usuário

Duas expressões comumente usadas no teste de usabilidade são usabilidade e experiência do usuário. De acordo com Nielsen [Nielsen 1993], usabilidade está associada a cinco atributos, os quais são:

- **Facilidade de aprendizagem:** o sistema deve ser fácil de aprender de forma que o usuário possa começar a utilizá-lo rapidamente.
- **Eficiência:** o sistema deve ser eficiente, de modo que uma vez aprendido pelo usuário, ele possa atingir uma alta produtividade;
- **Facilidade de memorizar:** o sistema precisa ser facilmente memorizado, de forma que um usuário ocasional não tenha novamente que aprendê-lo.
- **Erros:** o sistema deve ser consistente durante o uso, ele deve possuir uma baixa taxa de erros e deve ter a possibilidade de se recuperar quando um erro acontece.
- **Satisfação subjetiva:** O sistema deve ser agradável para o uso de modo que o usuário se sinta satisfeito ao usá-lo.

Observa-se que nem sempre usabilidade pode significar facilidade de uso, pois ela também é relacionada a outros fatores, o que nos leva a considerar que um sistema pode permitir uma grande produtividade por partes dos usuários e não ser nenhum pouco fácil de usar, mas mesmo assim manter uma boa usabilidade. Por causa disso, o conceito de usabilidade pode ser relacionado ao termo “adequação ao uso”.

Para a compreensão de como a experiência do usuário se relaciona com usabilidade, [McNamara e Kirakowski 2006] apresenta os três elementos básicos que devem ser usados ao avaliar a tecnologia móvel: funcionalidade, usabilidade e experiência do usuário.

Funcionalidade se refere ao produto. A utilidade das características dos dispositivos, manutenibilidade e confiabilidade devem ser tratadas. Avaliar a funcionalidade de um produto é responder a seguinte questão: O que o produto faz?

Já usabilidade é a interação entre o usuário e o produto. Ela está relacionada ao usuário e por esse motivo o teste de usabilidade necessita ser feito com usuários reais. A questão avaliada é: O produto faz o que eu espero que faça?

A experiência do usuário é uma relação mais abrangente entre o produto e o usuário e investiga a experiência pessoal do usuário usá-lo. As questões para avaliá-la são: Como o usuário se sente? O que significa para ele? A experiência de uso foi importante e confortável?

Os três conceitos apresentados acima não são independentes e de alguma forma podem influenciar em um aspecto avaliado por outro. Em relação à usabilidade e experiência do usuário, uma má usabilidade contribui para uma experiência negativa do usuário, o que pode desencorajar o usuário a usar mais vezes a aplicação ou até mesmo não usá-la mais [McNamara e Kirakowski 2006]. Dessa forma, uma aplicação móvel com boa usabilidade proporciona uma boa experiência para o usuário que pode ser avaliada pelo teste de usabilidade.

3.5 Teste de usabilidade nas aplicações móveis

Os dispositivos móveis são utilizados para tarefas rápidas e oferecem informações para a interação do usuário com o ambiente. A interface de uma aplicação que é adequada para um computador de mesa pode não ser apropriada para um dispositivo móvel, pois o usuário móvel tem menor capacidade de processar e absorver conteúdo que um usuário que está sentado em frente a um computador de mesa [Betiol 2004]. Dessa forma, o ambiente móvel influencia nas características que a aplicação deve ter em relação à sua usabilidade.

Um fator importante observado no ambiente móvel é a “usabilidade imediata” que é importante para os usuários que têm pouca ou nenhuma experiência no uso de computador [Mackenzie and Sukoreff 2002]. Por exemplo, o usuário pode achar o dispositivo móvel complicado e não querer aprender formas mais complexas de interação com a aplicação móvel.

Nesse contexto, avaliação da usabilidade no ambiente móvel trata fatores que irão influenciar também na interação do usuário com o dispositivo móvel. Devem ser considerados fatores, tais como as necessidades e características do usuário móvel, o contexto de uso dos dispositivos móveis, as características das aplicações, além das limitações físicas dos dispositivos móveis (e.g. tela pequena e teclado limitado) que podem afetar a experiência do usuário móvel.

Em virtude disso, a interface de uma aplicação móvel é fundamental para a interação do usuário móvel com o dispositivo e não deve interferir nas tarefas que estão sendo desenvolvidas no ambiente. A interface deve possibilitar a comunicação entre o usuário e o dispositivo, e quanto maior for a sua usabilidade, mais fácil será essa comunicação [Ballard 2007].

Um dos instrumentos para avaliar a usabilidade da aplicação e a experiência do usuário móvel é o teste de usabilidade. De acordo com Ballard [Ballard 2007], o teste de usabilidade é uma técnica para assegurar a facilidade de uso de uma aplicação ao pedir para o usuário usar a

aplicação e observar como este se comporta em relação à mesma. Também pode se referir ao processo de envolver participantes que representam o público alvo para avaliar se o produto está e acordo com critérios específicos de usabilidade [Rubin 1994] [Weiss 2002];

O teste de usabilidade pode ser feito na fase de design e implementação com a finalidade de encontrar problemas de usabilidade da aplicação a tempo de consertá-las. No entanto, em [Nokia 2009] inclui uma fase de especificação de interface do usuário (UI Specification) no modelo de processo de desenvolvimento que contém todos os detalhes e descreve como os requisitos serão implementados na prática, além de definir como os usuários irão interagir com a aplicação móvel.

Além disso, os tipos de dispositivos, o ambiente de teste (e.g., emulador ou dispositivo, laboratório ou campo), a interface ou domínio das aplicações móveis e mobilidade de usuário discutidos na Seção 3.2 são fatores importantes que influenciam no teste de usabilidade.

Os diferentes tipos de dispositivos afetam a experiência do usuário porque cada dispositivo pode exibir a aplicação móvel de uma maneira diferente e dessa forma, a interface das aplicações pode não ser visualizada de maneira intuitiva e apresentar uma navegação confusa entre as suas telas.

Em relação ao ambiente de teste, no emulador é possível testar funcionalidades relacionadas à usabilidade, entretanto, questões, tais como tamanho da tela, estrutura do menu da aplicação no dispositivo móvel e sua cor só podem ser avaliadas no próprio dispositivo móvel. Além disso, as avaliações de usabilidade feita o teste em laboratório pode não extrair a informação mais importante a respeito da usabilidade da aplicação. Por exemplo, interrupções, movimento físico do usuário e barulho não são avaliados no teste de usabilidade em laboratório. Dessa forma, as avaliações são mais adequadas quando realizadas em campo, pois o teste feito em laboratório não simula o contexto no qual os dispositivos móveis são usados. No entanto, segundo Kjeldskov e Graham [Kjeldskov e Graham, 2003], 71% dos testes de usabilidade para dispositivos móveis são feitos em laboratórios. Isso porque o teste feito em campo requer um esforço extra da equipe de teste, tais como tempo e custo.

[Betiol 2004] apresenta um estudo comparativo que investiga qual a influência do ambiente móvel na avaliação dos resultados de usabilidade da interface com usuário de uma aplicação de Internet móvel em um telefone celular. Essa avaliação é feita através da técnica chamada de “ensaios de interação” que utilizou as três abordagens seguintes: (i) dentro do laboratório utilizando emulador; (ii) dentro do laboratório utilizando o telefone celular e a câmera de documentos e (iii) fora do laboratório utilizando o telefone celular e a minicâmera sem fio.

[Betiol 2004] constata que é possível avaliar a usabilidade utilizando um emulador no computador desde que o contexto de avaliação seja o mais próximo possível ao contexto de uso, além disso, também deve ser considerado a posição do usuário e a maneira como ele segura o dispositivo móvel, pois eles influenciam na interação do usuário com a aplicação móvel.

[Kaikkonen 2005] faz uma comparação do teste de usabilidade para aplicações móveis em laboratório e em campo. Esse trabalho mostra que as diferenças encontradas em ambos ambientes são poucas. Os problemas encontrados durante o teste em campo são: dificuldade de compreender a lógica da aplicação e o fato de que poucos usuários desempenham todas as tarefas enquanto estão em movimento. Além disso, o tempo de execução do teste em campo é bem mais demorado do que o teste em laboratório. No entanto, de acordo com Roto [Roto et al. 2004], o teste em campo é adequado não somente para avaliações de interação com um sistema, mas também para avaliar as situações do comportamento do usuário e o ambiente.

[Silveira et al. 2004] apresentam um trabalho que avalia a qualidade da interação de usuários com dispositivos móveis dentro de um ambiente real (campo). Essa avaliação é feita através do teste de usabilidade. Segundo [Silveira et al. 2004], no contexto de aplicações para dispositivo móveis, não é possível verificar a mesma quantidade de problemas como se o usuário estivesse interagindo com o dispositivo em um ambiente real, pois o fator da mobilidade influencia muito nesse tipo de teste.

Os trabalhos aqui discutidos apresentam resultados divergentes na avaliação da usabilidade em laboratório e em campo. No entanto, acredita-se que fazer com que o testador simule uma avaliação de usabilidade em laboratório utilizando um emulador próximo ao contexto de uso do dispositivo no ambiente móvel conforme apresentado em [Betiol 2004] depende muito do domínio e da interface da aplicação e não verifica fatores intrínsecos do dispositivo móvel (e.g., tela e teclado) que influenciam na usabilidade. Além disso, acredita-se que somente o teste em campo é capaz de avaliar fatores de usabilidade relacionados à mobilidade que podem afetar o comportamento do usuário ao interagir com a aplicação móvel.

3.6 Conclusão

Neste capítulo foi apresentada uma visão geral sobre teste de software mostrando os principais conceitos da área que serão utilizados no capítulo seguinte. O principal objetivo deste capítulo foi mostrar em quais aspectos do ambiente móvel, já contextualizado no Capítulo 2, o teste de aplicações móveis se diferencia do teste de aplicações *desktop*.

Como foi visto nas seções anteriores, existe uma preocupação na literatura em como avaliar a usabilidade da aplicação móvel tanto em laboratório quanto em campo. Com o intuito de melhorar essa avaliação são utilizadas diferentes técnicas no teste de usabilidade para capturar quais fatores do ambiente móvel podem ser testados em cada um dos ambientes.

No Capítulo 4 são apresentados os requisitos elicitados para aplicações móveis que atuam no processo de teste e na usabilidade das aplicações móveis de uma forma geral. Os requisitos podem ser adicionados ao processo de teste tradicional com o objetivo de suprir as questões intrínsecas do ambiente móvel não tratadas por aquele.

Capítulo 4

Requisitos para Testes

Este capítulo apresenta o conjunto de requisitos para o teste de aplicações móveis elicitados nesta dissertação, os resultados das entrevistas e as principais contribuições desta dissertação. A Seção 4.1 apresenta a metodologia utilizada para a realização deste trabalho. Na Seção 4.2, os requisitos propostos são definidos e classificados. Na subseção 4.2.1 os requisitos para o processo de teste de aplicações móveis são apresentados, sendo que na subseção 4.2.2 são apresentados os requisitos para o teste de usabilidade de aplicações móveis. Na Seção 4.3 é feita uma análise dos resultados obtidos nas entrevistas com os profissionais da área e um resumo dos requisitos propostos. Por fim, na Seção 4.4 são apresentadas as principais conclusões e considerações deste capítulo.

4.1 Metodologia

O desenvolvimento deste trabalho está organizado em quatro fases: (i) Levantamento teórico-prático; (ii) Coleta dos dados; (iii) Análise dos dados e (iv) Validação. Na fase do Levantamento teórico-prático, é realizado um extenso levantamento bibliográfico e são identificadas boas práticas utilizadas nas empresas para investigar como as aplicações para dispositivos móveis são testadas. Na fase de Coleta dos dados, com base no estudo realizado na primeira fase, dois questionários são elaborados e enviados para empresas selecionadas para coletar informações sobre as atividades de teste para aplicações móveis. Na fase de Análise dos dados, os dados coletados na segunda fase são analisados com o objetivo de determinar quais requisitos são pertinentes para o processo de teste de aplicações móveis. Com base na análise realizada, os requisitos são classificados e catalogados. Na fase de Validação, três aplicações móveis são testadas utilizando os requisitos elicitados com o objetivo de validar o benefício e aumento da qualidade dos testes com o uso dos mesmos, além disso, os requisitos propostos são utilizados durante o desenvolvimento de aplicações móveis na disciplina de Engenharia de Software (ES) 2009.1 ofertada pelo Departamento de Computação (DC) da Universidade Federal do Ceará (UFC). As fases desta metodologia são detalhadas nas próximas subseções.

4.1.1 Levantamento teórico-prático

O levantamento teórico-prático consistiu de uma extensa revisão na literatura, entrevistas com profissionais experientes no desenvolvimento e teste de aplicações móveis e análise de documentos de projetos relacionados com a área. O levantamento bibliográfico também foi feito através de guias para desenvolvimento de aplicações móveis disponibilizados na Internet por empresas especializadas nesta tecnologia (i.e., Bug Huntress [BugHuntress 2009], LioNBRIDGE [LioNBRIDGE 2009], Little Spring [LittleSpring 2009], Motorola [Motodev 2009], Nokia [Nokia 2009], Nokia Research Center [NokiaResearch 2009], Serco [Serco 2009], Smartphonemag [Smartphonemag 2009], Sony Ericsson [SonyEricsson 2009], TechRepublic [TechRepublic 2009], Test Quest [TestQuest 2009], WedCredible [WedCredible 2009] e W3C [W3C 2009]).

4.1.2 Coleta de dados

Após a pesquisa inicial, descrita na fase anterior, são elaborados dois questionários, um voltado para desenvolvedores e outro específico para testadores, ambos com experiência em aplicações móveis. A primeira versão dos questionários foi elaborada em novembro de 2008 com base nos requisitos para a construção de questionários de boa qualidade apresentado no estudo realizado por Bastos [Bastos et al. 2007] e foi revisada por um profissional de Qualidade de Software com conhecimento em aplicações móveis.

O principal objetivo dos questionários é identificar como o processo de teste de aplicações móveis é realizado nas empresas, além de verificar quais requisitos elicitados estão presentes nos processos adotados. Além disso, a aplicação dos questionários tem como objetivos específicos: (i) comparar os dados obtidos na literatura com a prática profissional do desenvolvedor e do testador; (ii) capturar como os desenvolvedores atuam quando se trata do desenvolvimento (incluindo aqui a etapa de teste) de aplicações móveis; e (iii) verificar se os mesmos se as atividades realizadas na prática pelos desenvolvedores complementam os testes durante esse processo.

A estratégia de elaborar dois tipos de questionários se deve ao fato de que, em geral, o desenvolvedor também testa as aplicações móveis durante o desenvolvimento. Os dois questionários são compostos de perguntas objetivas e subjetivas que possuem basicamente o mesmo conteúdo. A diferença entre os questionários são os tipos de perguntas que são direcionadas para o perfil do profissional (e.g., testador ou desenvolvedor).

Por exemplo, uma questão aberta direcionada ao testador é se existe algum processo de teste específico para aplicações móveis na empresa em que ele trabalha, e se sim, qual será o processo e a sua principal diferença para o processo de teste de aplicações tradicionais.

Os requisitos são prontamente identificados através de questões subjetivas, tais como, qual o tipo de processo de desenvolvimento utilizado para desenvolver aplicações móveis. Outros requisitos exigiam uma análise mais complexa e, nesse caso, as perguntas subjetivas ajudaram na coleta dos dados.

Dois critérios são utilizados para seleção dos entrevistados: ter participado de no mínimo um projeto de desenvolvimento de aplicações móveis (e.g., corporativa, jogo, e-commerce, site, social, educativa, utilitário e manual) e ter experiência de no mínimo três meses na área.

A Figura 4.1 apresenta os principais tipos de aplicações para dispositivos móveis apresentados nos questionários. De acordo com a Figura 4.1, verifica-se que a maioria dos entrevistados (aproximadamente 60%) tem experiência em aplicações móveis do tipo Corporativa.

A Figura 4.2 ilustra o perfil dos entrevistados durante a fase de coleta dos dados. Analisando a Figura 4.2, verifica-se que quanto ao perfil dos entrevistados, 23% desempenham o papel de testador, enquanto 46% são desenvolvedores. Essa amostra significativa de desenvolvedores se deve ao fato de que dependendo da empresa eles, além de desenvolverem as aplicações móveis, também realizam os testes.

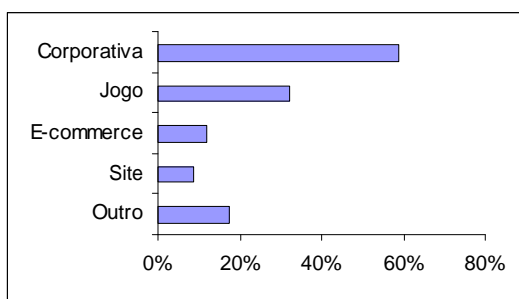


Figura 4.1 - Tipos de aplicações móveis

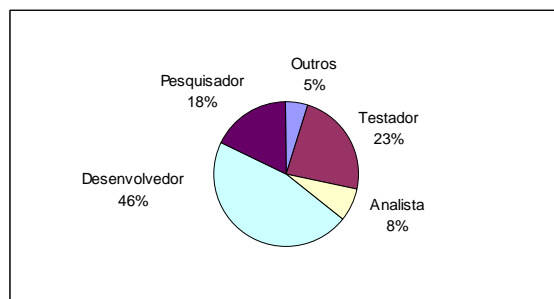


Figura 4.2- Perfil dos entrevistados

Os questionários foram disponibilizados *online* e o seu *link* foi direcionado aos profissionais de diversas empresas selecionadas. Contudo, onze questionários foram entregues pessoalmente durante visitas realizadas nas empresas. No entanto, levando em consideração o conteúdo do questionário, não houve indícios de que a sua forma de entrega impactou na confiabilidade dos

resultados obtidos. Ao todo foram quarenta profissionais de quinze empresas diferentes que contribuíram para essa pesquisa.

4.1.3 Análise dos dados

Nesta fase, os dados coletados durante as entrevistas foram analisados com o intuito de se obter um consenso de quais requisitos são testados durante o processo de teste de aplicações móveis. Essa análise consistiu no levantamento da porcentagem de uso dos requisitos propostos pelos profissionais entrevistados, além de investigar como os testes eram feitos de acordo com as limitações do contexto móvel. Dessa forma, os requisitos foram elicitados e classificados de forma genérica independente do tipo de aplicação móvel.

4.1.4 Validação

Com o intuito de validar a utilidade dos requisitos elicitados no teste de aplicações móveis, os mesmos são utilizados por dois testadores em três aplicações móveis desenvolvidas durante a disciplina de ES no semestre de 2006.1 pelos alunos do Curso de Ciência da Computação da UFC. Os requisitos propostos também são utilizados durante o desenvolvimento de aplicações móveis pelos alunos de graduação e pós-graduação na disciplina de ES no semestre 2009.1 ofertada pelo DC UFC.

Os testes das três aplicações móveis são realizados através da execução dos casos de testes por dois testadores, que não participaram do desenvolvimento dessas aplicações, utilizando os requisitos propostos. Os relatórios de teste gerados a partir do teste das três aplicações são analisados com o intuito de verificar como os requisitos elicitados podem auxiliar as equipes de teste de aplicações móveis a identificar erros inerentes do ambiente móvel.

Na disciplina de ES no semestre 2009.1, cada equipe que é composta, em média, por quatro alunos que recebem um modelo de relatório de teste no início do desenvolvimento do projeto de aplicações móveis. A estratégia é realizar os testes das aplicações móveis no emulador específico e no dispositivo móvel alvo³. Ao final, cada equipe deve gerar dois relatórios reportando os erros encontrados de acordo com os requisitos propostos nesta dissertação, além das diferenças encontradas nos dois ambientes de teste (i.e., emulador e dispositivo).

4.2 Requisitos Propostos

Os requisitos propostos nesta dissertação são obtidos a partir dos resultados da pesquisa

³ Os dispositivos móveis alvo são aqueles dispositivos para o qual a aplicação móvel terá como plataforma alvo.

realizada e representam boas práticas adotadas no processo de teste de aplicações móveis. Eles são critérios a serem seguidos pela equipe de teste com a finalidade de suprir as necessidades durante os testes das aplicações móveis. A documentação desses requisitos pode ser utilizada como um *checklist* dos itens que devem ser testados para auxiliar a equipe em suas atividades.

Os requisitos propostos nesta dissertação são documentados com o intuito de ajudar os processos de teste existentes a tratar questões do ambiente móvel de forma os testadores ao utilizar os requisitos elicitados possam identificar erros inerentes do ambiente móvel e, dessa forma, melhorar a qualidade das aplicações móveis.

Como mencionado no Capítulo 2, além das limitações das aplicações móveis, os testes de aplicações móveis devem tratar as limitações do ambiente *desktop*. Dessa forma, alguns dos requisitos propostos podem parecer intuitivos para um processo de teste. No entanto, não foi encontrado na literatura, um processo de teste que aborde as questões específicas do ambiente móvel.

Os requisitos propostos foram classificados em duas categorias: (i) Requisitos para o Processo de Teste de Aplicações Móveis; e (ii) Requisitos para o Teste de Usabilidade de Aplicações Móveis. A primeira categoria contém os requisitos relacionados ao processo de teste de aplicações móveis de uma forma geral. A segunda categoria contém os requisitos relacionados com aplicações móveis com o foco em usabilidade.

Nas próximas subseções são descritos os requisitos elicitados e é apresentado um resumo dos resultados das entrevistas em relação a cada requisito, mostrando um cenário atual dos testes de aplicações móveis.

4.2.1 Requisitos para o Processo de Teste de Aplicações Móveis

Requisito 1: O modelo de processo de desenvolvimento das aplicações móveis deve focar no processo de teste.

O processo de desenvolvimento deve focar no processo de teste para que se possa testar parte das funcionalidades da aplicação tão logo seja possível. Para atingir esse propósito, muitos fabricantes e desenvolvedores adotam uma abordagem incremental para o desenvolvimento de aplicações móveis [LioNBRIDGE 2008]. A abordagem incremental tem se mostrado eficaz para aplicações móveis, principalmente as aplicações que possuem alta complexidade onde nem sempre é possível especificar totalmente os requisitos nas fases iniciais do projeto. Dessa forma, esse modelo se torna apropriado para esse tipo de aplicação que pode tratar as limitações da tecnologia durante todo o desenvolvimento.

Na prática, a pesquisa realizada neste trabalho mostrou que 70% dos profissionais utilizam o modelo de processo incremental para o desenvolvimento de aplicações móveis. Uma empresa relatou que utiliza o modelo incremental em conjunto com o modelo de prototipação, pois considera mais eficaz para a validação dos requisitos com o cliente. Entretanto, 85% dos entrevistados não utilizam um modelo de processo de teste específico para aplicações móveis.

Embora algumas questões, tais como limitações da tecnologia móvel, heterogeneidade de hardware e software, testes em diferentes ambientes (e.g., emuladores e dispositivos), testes no próprio dispositivo e interrupções sejam tratadas, essa pesquisa não encontrou testadores usando um processo de teste específico para aplicações móveis.

Requisito 2: As aplicações devem ser testadas tanto em emuladores quanto em dispositivos móveis.

A maioria das aplicações móveis tem seu ambiente de desenvolvimento feito no computador e não no próprio dispositivo e, além disso, se comportam de diferentes maneiras dependendo do tipo de emulador utilizado e das versões de hardware e software do dispositivo a ser testado. Dessa forma, as aplicações devem ser testadas em dois ambientes: no emulador que executa no *desktop* e no próprio dispositivo móvel.

O teste feito no emulador deve ser usado pelos desenvolvedores como parte do seu processo de teste unitário. Existem muitos tipos de emuladores, tanto genéricos quanto os fornecidos pelo fabricante (também conhecidos como emuladores específicos), porém, na prática, muitos emuladores se comportam de diferentes maneiras para os dispositivos que eles simulam. Dessa forma, é apropriado que seja utilizado um emulador específico para o modelo do dispositivo alvo a ser testado. Nesse caso, o emulador específico é útil para validar a funcionalidade e compatibilidade das aplicações em ambiente controlado particularmente durante o seu ciclo de desenvolvimento e no teste de caixa branca. Além disso, o desenvolvedor e/ou testador estará trabalhando sob condições mais próximas da plataforma do dispositivo móvel [TestQuest 2004].

Contudo, os emuladores simulam apenas um subconjunto de variáveis do mundo real. No pior cenário, uma aplicação móvel pode até mesmo parar de funcionar quando transferida para o dispositivo [Legget et al. 2006]. Isso se deve ao fato de que testar as aplicações móveis no emulador utilizando um computador desktop nos faz esquecer-se da sua plataforma alvo (e.g., celular, PDAs ou outro dispositivo), na qual as aplicações móveis podem se comportar de maneiras diferentes.

Nesse contexto, o emulador falha em simular questões específicas do dispositivo móvel,

tais como tamanho de tela, diferenças de navegação, tipos de texto/ fonte, além de muitos bugs, gráficos (com renderização da cor), comunicação, recursos (bluetooth, IrDA, bateria ou captura de imagem), mobilidade e interoperabilidade da rede. Por exemplo, o teste de desempenho realizado no emulador pode não refletir o desempenho real do dispositivo por não considerar as características de desempenho da rede.

Portanto, o ideal é que os testes sejam feitos nos dois ambientes: emulador específico para o dispositivo alvo e no próprio dispositivo. A Tabela 4.1 apresenta as características específicas de aplicações móveis que devem ser testadas no emulador e no dispositivo.

Tabela 4.1 – Características testadas no emulador e no dispositivo móvel

Emulador	Dispositivo Móvel
Funcionalidades	Variações do dispositivo: tela, teclado, navegação
Usabilidade: <i>design</i> da <i>interface</i> da aplicação (e.g., <i>layout</i> e estruturas do menu)	Questões de usabilidade na navegação do dispositivo móvel
Comportamento do usuário (e.g., número de cliques no botão)	Desempenho, comunicação, recursos, mobilidade e interrupções do dispositivo e do mundo real

A pesquisa realizada mostrou que 73% dos profissionais realizam testes nos dois tipos de emuladores, dos quais 36% são feitos tanto em emulador genérico quanto em específico. Apenas 24% realizam os testes apenas em emulador genérico. Finalmente, 97% dos entrevistados realizam testes tanto em emulador quanto em dispositivo móvel. Os profissionais afirmam que testar no emulador ajuda a verificar as funcionalidades da aplicação e, dessa forma, alguns defeitos podem ser logo corrigidos economizando tempo e custo.

Embora os desenvolvedores estejam mais propensos a realizar os testes nos emuladores, constatamos que em alguns tipos de testes, tais como desempenho eles preferem realizar diretamente no dispositivo móvel conforme observado na literatura. Outro fator identificado foi que as equipes de testes preferem realizar logo os testes nos dispositivos alvos caso eles já estejam disponíveis para os testes.

Em suma, o que se tem feito na prática é testar as aplicações em emuladores específicos e dispositivos alvos. Quando não existe um dispositivo alvo para a aplicação, os testes primeiramente são feitos em emuladores genéricos (e.g., Sun Wireless Toolkit - WTK) e/ou emuladores específicos da própria plataforma (e.g., Symbian, Windows Mobile, JME e Android) e depois com modelos de dispositivos que possuam características distintas.

No entanto, isso não garante que a aplicação se comporte da mesma forma em todos dispositivos devido à diversidade de modelos, assim o ideal é que os testadores definam um

conjunto de dispositivos alvos para testar a aplicação móvel.

Requisito 3: O relatório de teste deve informar o nome da aplicação, a versão da aplicação, a versão do emulador e/ou dispositivo e o ambiente de teste.

O relatório de teste deve informar a versão do emulador e/ou dispositivo e o ambiente de teste além do nome e versão da aplicação (prática comum no teste de aplicações do tipo *desktop*), pois os resultados dos casos de testes podem divergir dependendo do contexto em que são realizados. Por exemplo, o teste realizado em um mesmo emulador pode apresentar resultados diferentes que depende do sistema operacional utilizado.

Na prática, 43% dos profissionais informam a versão do emulador e/ou dispositivo e 72% documentam o ambiente de teste. Dois testadores consideram importante também relatar o tempo de realização dos testes.

Requisito 4: Para cada erro reportado devem ser fornecidos a sua descrição, a sua frequência de ocorrência (sistemático, aleatório ou apenas uma vez), a sua localização na aplicação e um passo a passo para reproduzi-lo.

Os testadores devem reportar o erro de uma maneira que o desenvolvedor possa compreendê-lo e consertá-lo. Dessa forma, cada erro identificado deve ser apropriadamente registrado através da sua descrição, frequência, localização e, principalmente um passo a passo para ajudar o desenvolvedor a solucionar o problema de forma mais rápida e eficaz.

Esta pesquisa identificou que somente 22% dos profissionais informam todas as informações. No entanto, conforme ilustrado na Figura 4.3, 100% dos entrevistados fornecem a descrição do erro e 67% um passo a passo de como ele ocorre. Também foi constatado nas entrevistas que outras informações, tais como imagens do erro na aplicação e o grau de importância do erro também são informados pelos testadores.

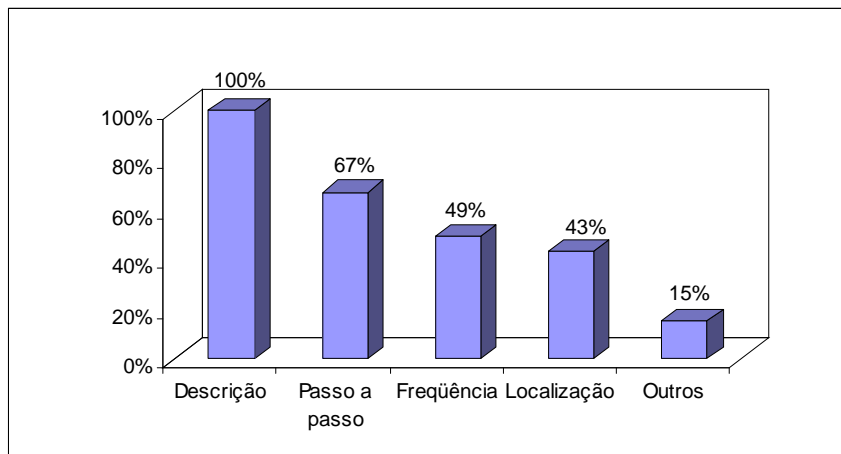


Figura 4.3 - Informações sobre o erro reportado

Requisito 5: As aplicações móveis devem ser testadas de acordo com as limitações do contexto móvel as quais elas se aplicam.

Os testadores devem testar as limitações específicas da tecnologia e que independem do tipo de aplicação, tais como, memória, processamento, tela, bateria, capacidade de armazenamento e mobilidade. Já outros fatores, como por exemplo, largura de banda depende do tipo de aplicação a ser desenvolvida. A seguir apresentaremos brevemente como alguns testes são realizados na prática conforme as limitações citadas.

Memória: nesse teste é inserida uma grande quantidade de dados (e.g., arquivos e imagens muito grandes) na aplicação. A quantidade de memória consumida é exibida no monitor de memória do emulador ou no dispositivo à medida que os dados da aplicação são manipulados. A experiência mostra que testar a memória no emulador não é confiável, pois o mesmo também utiliza a memória do computador. Dessa forma, o teste no dispositivo é mais seguro, pois possibilita verificar o uso de memória real da aplicação. O teste de memória também pode ser automatizado. Por exemplo, a ferramenta para automatizar o teste de memória, Device Explorer [Sony Ericsson developer 2009], faz o teste direto no dispositivo móvel via cabo USB. Embora possa ocorrer atraso devido à comunicação, esse teste é bem próximo do real. Nesse teste também é verificado a quantidade de memória que a aplicação ocupa no dispositivo após a sua instalação e se aplicável instalar e executar a aplicação a partir da memória externa (e.g., cartão de memória) do dispositivo.

Processamento: no teste de processamento são verificados os picos de processamento e travamento parcial ou total do aplicativo. A aplicação é inserida no emulador ou dispositivo e observa-se o tempo de resposta da aplicação, por exemplo, o tempo para abrir e executar uma

tarefa é aceitável e não compromete a funcionalidade da aplicação. Nesse teste também é medido o tempo de inicialização da aplicação, o tempo aceitável é em torno de 2 a 5 segundos e pode variar de acordo com o tipo de aplicação, por exemplo, algumas aplicações gráficas 3D podem apresentar um tempo de inicialização superior a 5 segundos. No caso de jogos móveis, o processamento é medido através do número de quadros por segundo. Embora o teste de memória possa ser feito no emulador, o ideal é que seja feito no dispositivo móvel alvo, e se houver mais de um dispositivo alvo o teste deve ser feito para cada um deles, pois a mesma aplicação pode apresentar comportamentos diferentes dependendo do dispositivo.

Tela: nesse teste é observada a visualização da aplicação em emuladores e/ ou dispositivos com resoluções e tamanhos diferentes. Como os dispositivos móveis apresentam telas pequenas e existem vários modelos com diferentes tipos resoluções, é necessário utilizar os dispositivos alvos para que se delimite o escopo e dessa forma os testes possam ser eficazes. Esse teste também pode ser feito através de protótipos nas fases iniciais de desenvolvimento e questões de design podem ser logo tratadas.

Bateria: nesse teste é avaliada a autonomia da bateria durante o uso da aplicação. Coloca-se a aplicação para rodar no dispositivo móvel usando todos os seus recursos (e.g., GPS e Bluetooth) e verifica-se se a quantidade de bateria consumida é aceitável. Também é feito o teste em relação armazenamento dos dados da aplicação caso a bateria seja retirada ou descarregada.

Capacidade de armazenamento: para testar a capacidade de armazenamento de um dispositivo móvel uma quantidade de dados é inserida na aplicação até que sua capacidade disponível esteja completamente cheia e verifica-se como a aplicação se comporta nesse cenário. Dessa forma, é testado se a aplicação continuará funcionando normalmente, como também o funcionamento das aplicações embarcadas (e.g., utilitários/serviços).

Mobilidade: esse teste pode ser feito em um ambiente controlado, porém o ideal é que seja feito no mundo real, com usuário em movimento e passível de interrupções para que o testador possa verificar como o fator mobilidade influencia no comportamento da aplicação.

Largura de banda: este teste depende do tipo de aplicação, se ela utiliza ou não tal recurso. Geralmente, o teste é feito através da transferência de dados pela rede (e.g., General Packet Radio Service - GPRS) para verificar o seu desempenho em relação a sua comunicação com a rede. Em alguns casos, as aplicações são testadas em laboratório usando Wi-Fi para fazer um comparativo em relação à taxa de transmissão do Wi-Fi com a rede. Entretanto, a experiência mostra que o comportamento da aplicação utilizando a banda geralmente é inferior, principalmente porque depende da localização do usuário móvel.

A Tabela 4.2 apresenta um resumo de como as restrições descritas acima são testadas na prática e os tipos de teste realizados pelos desenvolvedores e/ou testadores para testá-las.

Tabela 4.2 – Limitações testadas na prática nas aplicações para dispositivos móveis

Limitação	% de realização	Onde é feito	Como é feito	Tipo de Teste
Memória	80%	Emulador ou Dispositivo	Inserir uma grande quantidade de dados	Estresse, Carga e Desempenho
Processamento	63%	Emulador ou Dispositivo	Colocar a aplicação para abrir e executar uma tarefa	Estresse, Carga e Desempenho
Tela	40%	Emulador ou Dispositivo	Visualizar a aplicação em diferentes resoluções de tela	Usabilidade
Bateria	12%	Dispositivo	Colocar a aplicação para executar utilizando todos os seus recursos	Estresse
Capacidade de Armazenamento	20%	Dispositivo	Inserir uma quantidade de dados até a sua capacidade de armazenamento máxima	Estresse
Largura de Banda	12%	Dispositivo	Transferência de dados pela rede	Estresse, Desempenho e Interoperabilidade
Mobilidade	17%	Dispositivo	A aplicação é testada em situações reais de uso	Usabilidade, Estresse, Desempenho e Integração

Requisito 6: As aplicações móveis desenvolvidas não devem destruir as funcionalidades das aplicações já inseridas no dispositivo móvel.

Uma vez que uma aplicação é carregada no dispositivo, ela deve ser capaz de parar e retomar a sua execução quando o mesmo for interrompido por outros aplicativos [Nokia 2007]. Esses aplicativos podem ser interrupções causadas por serviços do próprio dispositivo, tais como chamada telefônica, mensagens SMS/ MMS, notificação de alarme, dentre outros ou por outras aplicações que rodam no dispositivo.

Na pesquisa realizada, 86% dos profissionais realizam esse teste e, se aplicável, também testam as interrupções na aplicação com outros tipos de recursos específicos do modelo de dispositivo móvel (e.g., Bluetooth).

Com esse tipo de teste pode-se também descobrir outros tipos de erros que nem sempre são causados pela aplicação. Por exemplo, uma equipe de teste relatou que uma aplicação pausava

corretamente durante o seu uso, porém quando ocorria uma interrupção (e.g. a chamada recebida) a aplicação não parava a sua execução. Nesse caso, os testes foram reportados e concluíram que o problema não era da aplicação e sim do próprio dispositivo alvo.

Requisito 7: O testador deve saber quais características podem ser testadas no laboratório e quais podem ser testadas no contexto móvel.

Da mesma forma que os testes das aplicações móveis devem ser feitos no emulador e no dispositivo, eles também devem ser realizados em laboratório e em campo.

No teste em laboratório, geralmente são verificadas as funcionalidades da aplicação e pode ser feito tanto em emulador quanto em dispositivo móvel. No entanto, o teste em campo é necessário para assegurar a própria validação de serviços e só pode ser desempenhada no próprio dispositivo em um ambiente controlado ou em “mundo real”.

Além disso, o teste em campo é mais adequado para avaliar o usuário móvel porque o contexto em que ele está inserido afeta o uso e o desempenho [Kaikkonen et al. 2005]. Dessa forma, se o teste em campo não é realizado os testadores deixam de verificar características que podem afetar o desempenho do usuário móvel, tais como interrupções, movimento, barulho, multitarefa, iluminação, dentre outras.

Os testadores devem compreender a natureza do contexto móvel e as necessidades do usuário móvel e também serem capazes de testar as aplicações em campo que é muito mais trabalhoso do que em laboratório. O ideal é que o testador descubra todas as situações que cada usuário passa no seu cotidiano, pois alguns cenários usados para a execução dos testes podem ser fora de contexto dos usuários alvos e falharem em tais situações [Roto et al. 2008]. Para viabilizar os cenários, os testadores devem conduzir os testes de aplicações móveis com usuários reais para identificar erros comuns e melhorar a usabilidade da aplicação. A Tabela 4.3 apresenta as características do teste feito em laboratório e em campo.

Na pesquisa realizada todas as aplicações são testadas em laboratórios e apenas 30% dos entrevistados só realizam o teste em campo quando é necessário testar algum recurso que só pode ser observado nesse ambiente (e.g., GPRS) ou também em alguns casos para verificar a usabilidade da aplicação. No entanto, 92% das aplicações são desenvolvidas com foco nos usuários reais e apenas 42% são testadas com eles. Na prática, o que ocorre é que quem faz o teste é o testador ou o desenvolvedor e, geralmente, não existe uma amostra de usuários alvos para realizá-lo.

Tabela 4.3 – Características do Teste em Laboratório e em Campo

Teste em Laboratório	Teste em Campo
Emulador e/ou dispositivo móvel	Dispositivo móvel
Ambiente controlado	Ambiente pode ser controlado e/ou não
Pode ser feito sentado de frente para o computador ou sentado segurando o dispositivo ou apoiado sobre a mesa	Pode ser feito em movimento, em pé/ sentado, em casa/escritório, com silêncio/barulho, no claro/escuro)
Adequado para aplicações que não necessitam se conectar a rede	Ideal para testar recursos específicos do contexto móvel (e.g., conectividade, mobilidade e interrupção)
Foca nos componentes de aplicações móveis (e.g., <i>layout</i> , apresentação da informação, menu) que não são influenciados pelo contexto móvel.	Adequado para estudar o comportamento das aplicações móveis no contexto móvel.

Requisito 8: O teste de usabilidade deve ser incluído durante o ciclo de desenvolvimento da aplicação móvel.

No ambiente móvel, o teste de usabilidade foca em fatores, tais como requisitos da aplicação que influenciam na sua utilização, nível de satisfação do usuário, facilidade de aprendizagem (e.g., navegação fácil e intuitiva), terminologia e informações da aplicação (e.g., mensagens e termos utilizados), capacidade (e.g., velocidade da aplicação), interface (e.g., qualidade das imagens, estrutura dos menus e cores utilizadas). Além desses fatores, o teste de usabilidade se preocupa principalmente, em como a aplicação se apresenta na pequena tela do dispositivo móvel. Para este fator são testadas características relacionadas à quantidade e a organização da informação na tela, as sequências das telas, os caracteres do texto da aplicação e sua forma (e.g., negrito, itálico, sublinhado, uso de cores e letras maiúsculas).

No entanto, a mobilidade do usuário e o ambiente influenciam no teste de usabilidade das aplicações móveis. Nesse contexto, o teste de usabilidade é um processo mandatório para assegurar que uma aplicação móvel seja prática, eficaz e fácil de usar, especialmente da perspectiva do usuário móvel [Overlander 2008] [Zhang e Adipat 2005] [Kjeldskov et al. 2004] [Kjeldskov e Skov 2003]. Por exemplo, o usuário pode estar em pé, andando ou sentado em ambiente escuro ou com muita luz enquanto usa a aplicação e ainda ter a possibilidade de realizar diversas atividades no seu dispositivo móvel. Consequentemente, estudos de campo são mais apropriados para o teste de usabilidade quando o desempenho da aplicação tem alta dependência do contexto móvel [Zhang e Adipat 2005].

Além disso, a equipe de teste deve compreender que o teste de usabilidade para o contexto móvel está longe de ser amarrar a um contexto específico e que alguns métodos

tradicionais de usabilidade não são diretamente aplicáveis a esse ambiente. Limitações, tais como telas pequenas, processadores mais lentos, falta de armazenamento de dados persistentes e medida de largura de banda inferior requerem métodos de testes diferentes. Dessa forma, testadores devem estar atentos para que os desenvolvedores usem métodos específicos para compensar essas restrições e assim desenvolver boas estratégias para testar esses métodos.

Na pesquisa realizada 56% dos profissionais realizam o teste de usabilidade nas aplicações móveis. No entanto, apenas 18% realizam esse teste em campo. Assim, características do ambiente que afetam a usabilidade de uma aplicação nem sempre são testados.

4.2.2 Requisitos para o Teste de Usabilidade de Aplicações Móveis

Os princípios de usabilidade testados nas aplicações de uma maneira geral também devem ser seguidos nos testes de aplicações móveis. Dessa forma, os requisitos apresentados nesta subseção são requisitos de usabilidade que se tornam mais críticos e até exclusivos do ambiente móvel e, portanto, fundamentais no teste de usabilidade das aplicações móveis.

Requisito 9: O usuário móvel deve sempre saber onde está na aplicação, o que ele já fez, o que pode fazer e como desfazer os seus erros.

Essa regra geral de usabilidade se aplica muito bem ao contexto móvel devido às restrições que a tecnologia apresenta (e.g., tela pequena com diferentes resoluções e teclado limitado). Dessa forma, cada tela de uma aplicação móvel deve informar ao usuário sobre a situação atual da aplicação e, para isso, a tela deve conter todas essas informações aproveitando ao máximo o seu tamanho. Por exemplo, a ausência de mensagens de retorno para o usuário impede que ele saiba que ação a aplicação está executando. O ideal é que haja um tratamento na aplicação sendo exibidas mensagens de aviso ou alerta para que o usuário saiba o que está acontecendo.

Outro problema básico de usabilidade com aplicações móveis é que os usuários gastam um tempo significativo navegando no conteúdo através de uma série de menus antes de obter o que eles querem, dessa forma, eles frequentemente perdem a estrutura do menu, especialmente se o usuário não é familiar com a aplicação. Portanto, os menus da aplicação devem ser fáceis de acessar e menus importantes não devem ficar escondidos, pressionando poucas teclas deve-se chegar a eles. Por exemplo, a configuração de uma aplicação deveria estar no menu principal e não dentro de um submenu.

Contudo, a pesquisa mostrou que apenas 44% dos profissionais testam este requisito.

Requisito 10: As ações do usuário móvel não devem ser desfeitas quando interrupções

externas forçam a aplicação pausar.

Quando um usuário entra com uma informação textual em uma tela pequena através de um teclado limitado, a aplicação não deve perdê-la durante uma interrupção (e.g., chamada recebida, mensagem - MMS/ SMS, Bluetooth, IrDa, alarme, remoção da bateria ou outro evento externo). Se os usuários se sentem frustrados em ter que digitar um texto, eles se sentirão mais ainda se tiverem que digitar novamente nas condições apresentadas. Dessa forma, a aplicação deve retomar o seu estado corretamente quando o usuário móvel é interrompido.

Por exemplo, para testar este requisito o testador poderá executar os seguintes passos:

- Inicia a aplicação móvel;
- Enquanto usa a aplicação receber uma chamada de outro dispositivo móvel; e
- Observa o comportamento da aplicação após o recebimento da chamada:
 - Testador atende a chamada e após seu término verifica o estado da aplicação.
 - Testador recusa a chamada e verifica o estado da aplicação.

Na pesquisa realizada, apenas 50% dos profissionais testam se a aplicação salva o seu estado atual durante uma interrupção.

Requisito 11: A aplicação deve suportar os formatos de tela retrato e paisagem.

Uma tendência atual é que os próprios dispositivos dêem o suporte para os dois formatos. No entanto, em alguns dispositivos (e.g., Nokia N95) o suporte é parcial, pois o usuário ao escolher o modo paisagem não consegue interagir com a aplicação através do teclado. Já em outros dispositivos depende da linguagem em que a aplicação é desenvolvida (e.g., JME ou Flash). Dessa forma, quando o dispositivo não é compatível com os dois formatos a aplicação terá que dar o suporte manualmente.

Nesse contexto, quando as aplicações móveis ajustam seus *layouts* para se adequar ao formato escolhido (e.g., retrato e paisagem), melhora a usabilidade e encoraja a interação do usuário. No teste de aplicações do tipo *desktop* esse requisito não é testado. Isso acontece porque seus monitores raramente são usados para girar 90 graus [BugHuntress 2007]. No teste de aplicações móveis, especialmente em jogos, é fundamental testar esse requisito.

Apesar de ser um requisito particular do contexto móvel, apenas 32% dos profissionais testam se a aplicação suporta os dois formatos de tela.

Requisito 12: As aplicações móveis devem pedir a permissão do usuário antes de

estabelecer conexões.

Custos incorridos pela aplicação sem o conhecimento do usuário podem levar a questões legais. Da mesma forma, quando a aplicação cobra pelos seus serviços, a mesma deve informar corretamente os custos associados ao uso do serviço.

No entanto, a solicitação de permissão para estabelecer conexões pode ser feita pela plataforma do dispositivo móvel, se a aplicação que for executada no mesmo não for certificada.

Na pesquisa realizada, 68% dos profissionais testam este requisito.

Requisito 13: As aplicações móveis não devem usar texto justificado, alinhado à direita, cortado, todo em maiúsculo, e sua quantidade deve ser mínima para que não seja necessário utilizar barra de rolagem. Além disso, a aplicação móvel deve ter um bom contraste em relação ao plano de fundo.

Uma das limitações do dispositivo móvel é a de possuir tela pequena. Para tratar essa limitação a apresentação da informação exibida na tela deve ser bem elaborada e não repetitiva para que não dificulte a visualização do usuário. Textos justificados em telas pequenas apresentam espaços grandes que interrompem os movimentos dos olhos e impedem a leitura e não deveriam ser utilizados.

Algumas vezes, os usuários não notam todas as informações nos itens ou listas, se elas não estão prontamente visíveis. Portanto, a informação mais importante de cada tela deve estar disponível sem a barra de rolagem. Entretanto, quando os textos são muito longos e não é possível evitar a barra de rolagem, o usuário não deveria ser forçado a rolar um texto longo linha por linha, para isso a velocidade da barra de rolagem deveria aumentar quando o usuário pressionasse para baixo um botão do dispositivo móvel associado a essa função. Além disso, a informação deve apresentar diferentes níveis de contraste para que as cores fiquem claramente distinguíveis e o usuário possa saber qual o foco atual da aplicação.

Se aplicável, os testadores devem tratar da mesma forma as imagens e as tabelas em relação à formatação e exibição. As imagens não devem ser distorcidas e/ou cortadas e devem ser redimensionadas para se adequar a resolução do dispositivo alvo.

Na pesquisa realizada, 44% dos profissionais testam se a aplicação móvel não usa texto justificado, 80% testam se ela apresenta um bom contraste, 82% testam se não utiliza barra de rolagem e 50% testam se o texto é alinhado à direita, cortado, todo em maiúsculo e se sua quantidade é mínima.

Requisito 14: As aplicações móveis devem requerer a confirmação do usuário para

executar ações não reversíveis.

O usuário móvel tem dificuldade de inserir dados em uma tela pequena e geralmente comete erros de digitação. Dessa forma, a aplicação deve solicitar a confirmação das ações que não são reversíveis. O ideal é que a aplicação corrija os dados do usuário quando este entra com dados no formato errado.

Na pesquisa realizada, 82% dos profissionais testam esse requisito.

Requisito 15: As aplicações móveis devem explorar a utilidade e a usabilidade proporcionadas pelo som durante a sua execução.

As aplicações móveis devem ser testadas com diferentes configurações de som e o usuário móvel deve ser capaz de controlar esses sons, ajustando o volume ou desligando completamente. Por exemplo, mensagens de *pop-up* podem emitir um tipo de som se o usuário tem tons de alerta de mensagens ligados no dispositivo [Mikhaleenko 2006]. Nessa situação muitos pop-ups irritariam o usuário móvel prejudicando a usabilidade da aplicação.

Os testadores devem ter isso em mente ao testarem aplicações móveis com dimensões audíveis, principalmente jogos móveis. Nesse caso, o ideal é que o jogo tenha uma música com volume baixo ao fundo e possa ter diferentes tipos de som, por exemplo, uma música para o menu principal, uma música para finalizar o jogo e outra música para os demais eventos.

De acordo com as entrevistas, 38% dos profissionais testam se os usuários podem controlar os sons da aplicação e apenas 14% testam a aplicação com diferentes configurações de som.

Requisito 16: As teclas do tipo *softkey* quando utilizadas para controlar as ações das aplicações móveis são utilizadas conforme projetadas para o dispositivo móvel.

Conforme ilustra a Figura 4.2, a variedade dos modelos de dispositivos em relação ao design acarreta em diferentes formas de acesso as funcionalidades da aplicação, tais como diferentes tipos de teclado (e.g., QWERT e numérico convencional), tela (sensível ao toque com ou sem a caneta), sensor de movimento, *jog dial*⁴, *track ball* e as teclas *softkeys*.

⁴ Um "Jog Dial" é um botão redondo rotativo que serve para ajustar/selecionar o volume do som.



Figura 4.4 – Variedade de modelos de dispositivos em relação ao design

A tecla *softkey* é um botão que desempenha uma ação que é mostrada próxima a ela na tela da aplicação. As teclas do tipo *softkey* podem ser do tipo *softkey* da esquerda (Left Soft Key – LSK), *softkey* da direita (Right Soft Key - RSK) e *softkey* do meio (Center Soft Key - CSK). Nesse contexto, elas devem mapear as ações mais importantes e mais usadas na aplicação móvel. Elas devem ser intuitivas e utilizadas na aplicação conforme são projetadas no dispositivo móvel. Na literatura recomenda-se utilizar a tecla *softkey* da esquerda para ações positivas (e.g., menu, selecionar e Ok) e a tecla *softkey* da direita para ações negativas (e.g., sair, voltar e cancelar). No entanto, nem todas as aplicações móveis as usam dessa maneira. Na maioria das vezes, a *softkey* do meio também é usada para ações positivas.

Além disso, o testador deve testar se as teclas *softkeys* funcionam conforme as ações especificadas nos seus rótulos. Por exemplo, o usuário ao clicar no comando “voltar” espera que a aplicação volte para a tela anterior. Seus rótulos também devem manter a mesma linguagem padrão, se o padrão é português, então todo menu da aplicação deve se apresentar nessa mesma língua.

Na pesquisa realizada 74% dos profissionais testam se as *softkeys* da esquerda e da direita são utilizadas pela aplicação conforme projetadas no dispositivo móvel. No entanto, 50% das aplicações móveis utilizam a *softkey* da direita para ações positivas e 53% utilizam a *softkey* da esquerda para ações negativas.

4.3 Resumo dos Requisitos Propostos

Na seção anterior descrevemos os requisitos propostos capturados da literatura e das entrevistas realizadas com profissionais experientes no desenvolvimento de aplicações móveis. Além disso, foi apresentado os resultados das entrevistas em relação à porcentagem do uso dos requisitos propostos na prática por profissionais que atuam na área. A Tabela 4.4 resume os requisitos para o processo de teste de aplicações móveis e a Tabela 4.5 resume os requisitos para o teste de usabilidade de aplicações móveis apresentados neste capítulo de acordo com a porcentagem de uso nas entrevistas.

Tabela 4.4– Requisitos para o Processo de Teste de Aplicações Móveis

Requisitos Propostos	Porcentagem de uso nas entrevistas	
Requisito 1: O modelo de processo de desenvolvimento deve focar no processo de testes.	70%	
Requisito 2: As aplicações móveis devem ser testadas tanto em emuladores quanto em dispositivos móveis	97%	
Requisito 3: O relatório de teste deve informar o nome da aplicação, a versão da aplicação, a versão do emulador e/ou dispositivo e ambiente de teste.	43%	
Requisito 4: Para cada erro reportado devem ser fornecidos a sua descrição, a sua frequência de ocorrência, a sua localização na aplicação e um passo a passo para reproduzi-lo.	22%	
Requisito 5: As aplicações móveis devem ser testadas de acordo com as limitações do contexto móvel as quais elas se aplicam.	Memória	80%
	Processamento	63%
	Tela	40%
	Bateria	12%
	Armazenamento	20%
	Largura de Banda	12%
	Mobilidade	17%
Requisito 6: As aplicações móveis desenvolvidas não devem destruir as funcionalidades das aplicações já inseridas no dispositivo móvel.	86%	
Requisito 7: O testador deve saber quais características podem ser testadas no laboratório e quais podem ser testadas no contexto móvel.	Laboratório	100%
	Campo	30%
Requisito 8: O teste de usabilidade deve ser incluído durante o ciclo de desenvolvimento da aplicação móvel.	56%	

Tabela 4.5 – Requisitos para o Teste de Usabilidade de Aplicações Móveis

Requisitos Propostos	Porcentagem de uso nas entrevistas	
Requisito 9: O usuário móvel deve sempre saber onde está na aplicação, o que ele já fez, o que pode fazer e como desfazer os seus erros.	44%	
Requisito 10: As ações do usuário móvel não devem ser desfeitas quando interrupções externas forcem a aplicação pausar.	50%	
Requisito 11: A aplicação deve suportar os formatos de tela retrato e paisagem.	32%	
Requisito 12: As aplicações móveis devem pedir a permissão do usuário antes de estabelecer conexões.	68%	
Requisito 13: As aplicações móveis não devem usar texto justificado, alinhado à direita, cortado, todo em maiúsculo, e sua quantidade deve ser mínima para que não fosse necessário utilizar barra de rolagem. Além disso, a aplicação móvel deve ter um bom contraste em relação ao plano de fundo.	Texto justificado	44%
	Alinhado à direita, cortado e todo em maiúsculo	50%
	Barra de rolagem	82%
	Bom contraste	80%
Requisito 14: As aplicações móveis devem requerer a confirmação do usuário para executar ações não reversíveis.	82%	
Requisito 15: As aplicações móveis devem explorar a utilidade e a usabilidade proporcionadas pelo som durante a sua execução.	Controle do som	38%
	Diferentes configurações	14%
Requisito 16: As teclas do tipo <i>softkey</i> quando utilizadas para controlar as ações das aplicações móveis são utilizadas conforme projetadas para o dispositivo móvel.	74%	

No entanto, além dos requisitos elicitados, foram identificados pontos relevantes durante a pesquisa. Primeiramente, foi observado que vários tipos de testes são realizados durante o processo de desenvolvimento de aplicações móveis, mas ainda não há um consenso entre a literatura e a prática, de qual conjunto de testes é o ideal para essas aplicações. O que se tem feito na prática é que ou se adota um conjunto de tipos de teste para qualquer tipo de aplicação móvel, e estes podem variar conforme solicitação do cliente e necessidades do usuário ou se adota um conjunto de tipos de testes que variam conforme o tipo de aplicação móvel.

Conforme ilustra a Figura 4.3, o Teste Funcional, o Teste de Unidade e o Teste de Usabilidade são os executados com maior frequência, independente do tipo de aplicação móvel. Já 88% dos entrevistados com o perfil de desenvolvedor realizam o teste de desempenho em aplicações móveis do tipo jogos e 55% dos desenvolvedores e/ou testadores realizam teste de sistema e teste de integração em aplicações móveis corporativas.

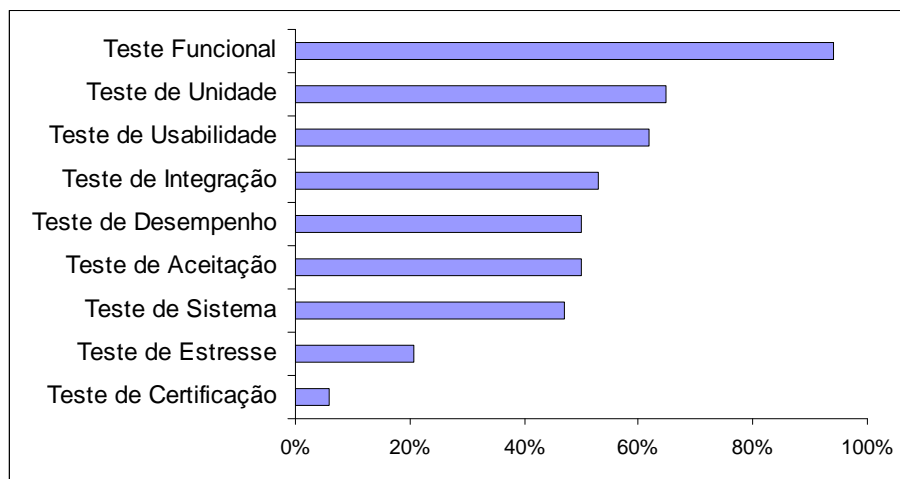


Figura 4.5 - Tipos de teste realizados na prática

Outro ponto constatado é que, embora existam diversas ferramentas (e.g., Device Explorer [Sony Ericsson developer 2009] e TestQuest Pro [TestQuest 2004]) disponíveis que automatizam alguns testes feitos no emulador e/ou dispositivo móvel, apenas 10% dos entrevistados realizam algum tipo de teste automatizado, tais como o teste de unidade e desempenho.

4.4 Conclusões

Este capítulo propôs um conjunto de requisitos elicitados através da investigação de como o processo de teste de aplicações móveis é realizado na literatura e na prática. Embora os requisitos apresentados não englobem todo o processo de teste, eles podem ser utilizados como base para auxiliar as equipes de teste nas atividades essenciais de teste de aplicações para dispositivos móveis.

É importante ressaltar que, nem todos os resultados obtidos nas entrevistas puderam ser definidos como requisitos para o teste de aplicações móveis (por exemplo, qual o conjunto de tipos de testes ideal para uma aplicação móvel), pois não foi possível evidenciar um consenso na literatura ou na prática.

Dentre as conclusões obtidas nas entrevistas, é importante destacar os seguintes pontos: seguir. Grande parte das empresas não possui um processo de teste específico para testar aplicações móveis, portanto, os testes são realizados de forma não sistemática, baseados, geralmente, na experiência ou intuição dos desenvolvedores e testadores. Outro fator observado é que com a ausência de requisitos nos processos de teste para aplicações móveis utilizados nas

empresas, aspectos importantes deixam de ser testados e erros triviais não são identificados.

O capítulo seguinte apresenta um estudo de caso que descreve os resultados obtidos na utilização dos requisitos propostos nos testes de três aplicações móveis e uma disciplina de Engenharia de Software do Departamento de Computação da UFC. Além disso, é feita uma análise dos resultados alcançados mostrando como os requisitos propostos ajudaram a identificar erros inerentes do ambiente móvel.

Capítulo 5

Validação

Este capítulo apresenta os resultados obtidos com o uso dos requisitos para o teste de três aplicações móveis e em uma disciplina de Engenharia de Software. A Seção 5.1 apresenta uma descrição das aplicações móveis utilizadas para a validação dos requisitos elicitados e na Seção 5.2 é descrito o ambiente de teste adotado. A Seção 5.3 apresenta os resultados obtidos com a utilização dos requisitos propostos nas aplicações Shopping Móvel (Seção 5.3.1), Museu Móvel (Seção 5.3.2) e Teatro Móvel (Seção 5.3.3), além de apresentar os resultados obtidos com o uso dos requisitos elicitados na disciplina de Engenharia de Software por alunos da graduação e pós-graduação (Seção 5.3.4). Na Seção 5.4 é apresentada uma discussão dos resultados obtidos, e por fim, a Seção 5.5 traz as considerações finais sobre este capítulo.

5.1 Estudo de Caso

Com o intuito de validar os benefícios obtidos com o uso dos requisitos propostos no Capítulo 4, os mesmos foram utilizados por dois testadores nos testes de três aplicações móveis desenvolvidas pelos alunos de graduação da disciplina de Engenharia de Software (ES) do Departamento de Computação da UFC no ano de 2006. Essas aplicações são: Shopping Móvel, Museu Móvel e Teatro Móvel. Além disso, os requisitos foram aplicados pelos próprios alunos, neste caso, de graduação e pós-graduação, durante o desenvolvimento de dez aplicações móveis na disciplina de Engenharia de Software de 2009.

O Shopping Móvel é uma aplicação cliente-servidor desenvolvida em JME que permite ao usuário visualizar o conteúdo de lojas e realizar compras em um shopping virtual por meio de dispositivos móveis. Além disso, a aplicação permite a visualização do estoque de lojas, a consulta de preços e, através de um perfil cadastrado pelo usuário, a sugestão de promoções.

O Museu Móvel é uma aplicação cliente-servidor desenvolvida em JME que funciona como um guia virtual, permitindo que os visitantes de um museu possam acessar, a partir do seu dispositivo móvel, informações sobre as obras. O usuário é interligado a rede sem fio interna do museu através de um acesso restrito.

O Teatro Móvel é uma aplicação cliente-servidor desenvolvida em JME que permite ao usuário realizar a compra de ingressos de um teatro através de um dispositivo móvel. A aplicação permite visualizar a programação da semana, comprar e reservar ingressos, consultar preços e, através de um perfil cadastrado pelo usuário, receber sugestões de peças para o usuário.

Após feita a análise dos resultados obtidos na utilização dos requisitos elicitados nessas três aplicações móveis, os requisitos propostos foram utilizados por dez equipes compostas por alunos de graduação e pós-graduação da UFC durante o desenvolvimento de um projeto na disciplina de Engenharia de Software no semestre de 2009.1, ofertada pelo Departamento de Computação da UFC. O projeto da disciplina de ES consiste no desenvolvimento de uma aplicação móvel por cada equipe que envolve um cliente e um servidor utilizando uma linguagem de programação escolhida pelos alunos [Lelli et al. 2008].

5.2 Ambiente de Teste

Conforme mencionado na seção anterior, as aplicações móveis foram desenvolvidas pelos alunos de graduação e pós-graduação da disciplina de Engenharia de Software do Departamento de Computação da Universidade Federal do Ceará em 2006 e em 2009. Para maiores detalhes sobre a metodologia utilizada nessa disciplina, o leitor deve consultar [Andrade et al. 2008].

Durante a disciplina de ES, os alunos são divididos em equipes para o desenvolvimento de aplicações móveis. Cada aplicação a ser desenvolvida é o projeto de disciplina e, em geral, ela envolve um cliente e um servidor, e, necessariamente, é gerenciado utilizando uma ferramenta de gerência de projetos, conforme apresentado em [Lelli et al. 2008]. O cliente é a parte da aplicação que executa em um dispositivo móvel, e o servidor, dependendo do tipo de aplicação a ser desenvolvida, pode ser executado em um dispositivo móvel ou em um computador de mesa.

Cada equipe desenvolve uma aplicação de acordo com um modelo de processo de desenvolvimento (e.g., cascata, espiral e incremental) que pode ser escolhido pela equipe ou designado pelo professor.

Além disso, independente do modelo de processo adotado pela equipe, a disciplina possui um conjunto de artefatos resultantes de cada etapa dos modelos de processo de desenvolvimento que devem ser entregues por todas as equipes. Um dos resultados exigidos é o conjunto de artefatos de Testes que é composto pelo Plano de teste e pelo Relatório de Teste. O Plano de Teste contém os casos de testes e o Relatório de Teste os erros encontrados durante a execução dos casos de teste.

O ambiente de teste utilizado pelos dois testadores durante os testes das três aplicações móveis e pelos próprios desenvolvedores (i.e., alunos de graduação e pós-graduação) nas dez aplicações citadas na Seção 5.1. é apresentado nas próximas subseções.

5.2.1 Shopping Móvel, Museu Móvel e Teatro Móvel

As aplicações Shopping Móvel, Museu Móvel e Teatro Móvel foram desenvolvidas por três equipes utilizando o modelo de processo de desenvolvimento Cascata na disciplina de ES em 2006. Cada equipe era composta, em média, por quatro alunos de graduação do Curso de Ciência da Computação da UFC. Apenas a equipe que desenvolveu a aplicação Shopping Móvel possuía integrantes com experiência no desenvolvimento de aplicações móveis.

Os testes das três aplicações móveis são realizados por dois testadores que não participaram do desenvolvimento das aplicações móveis. Um dos testadores é aluno de graduação do curso de Ciência da Computação da UFC com pouco conhecimento em testes, entretanto, já havia cursado a disciplina de ES do DC, participando do projeto de aplicações móveis desenvolvido durante a mesma. O outro testador é aluno do curso de mestrado do Mestrado e Doutorado em Ciência da Computação (MDCC) da UFC com bom conhecimento em testes e experiência no projeto de aplicações móveis.

Cada testador utiliza um modelo de relatório de teste que contém os requisitos propostos para serem seguidos durante a execução dos casos de teste. Estes foram elaborados pelas equipes que desenvolveram as aplicações móveis e são apresentados no anexo A desta dissertação.

Os testes são realizados em laboratório com o testador sentando em frente ao computador usando um emulador e em movimento usando um dispositivo móvel conectado na rede sem fio (i.e., Wi-Fi). Dessa forma, como os testes são feitos em laboratório usando apenas a conexão Wi-Fi, não são realizados testes para verificar o desempenho das aplicações móveis usando, por exemplo, a rede celular GSM através da conexão GPRS.

Além disso, os servidores das três aplicações foram desenvolvidos para executarem em um computador de mesa e são povoados com imagens de alta resolução (e.g., 1024 x 1024 pixels), muito texto e imagens com tamanho grande (e.g., 10MB).

O ambiente de execução dos testes é composto por:

- Um emulador genérico Sun® Wireless Toolkit versão 2.5.2 (WTK) [Sun microsystems 2009]; um emulador específico Nokia® S60 3rd Edition Feature Pack 1 (S60) [Forum.Nokia.com 2009] para o modelo de dispositivo alvo Nokia® N95;

um emulador específico Sony Ericsson® SDK JP8 2.5.0.3 (JP8) [Sony Ericsson developer 2009] para o modelo de dispositivo alvo Sony Ericsson® G705. Todos são executados no Windows® XP.

- Um celular Nokia® N95 (N95) com processador ARM11 330 MHz e um celular Sony Ericsson® G705 (G705) com processador ARM9 134 MHz. Ambos dispositivos possuem tela com resolução 240x320 pixels e conectividade sem fio Wi-Fi usando o padrão IEEE 802.11g/b.

A Tabela 5.1 resume as principais características dos celulares utilizados para o teste das três aplicações móveis.

Tabela 5.1 – Especificação dos celulares utilizados nos testes

Características	Nokia® N95	Sony Ericsson® G705
Sistema Operacional	Symbian 9.2	Proprietário Sony Ericsson Java Platform 8
Tela	2.6” Rotação automática 16 milhões de cores	2.4” Rotação automática 262.144 cores
Resolução	240 x 320 pixels	240 x 320 pixels
Armazenamento/ Memória	160MB de memória interna 64MB de SDRAM 18MB de memória livre executável 1 GB Flash ROM	120MB de memória interna 128MB de RAM
Conectividade	GSM/GPRS IEEE 802.11b/g Bluetooth USB IrDA	GSM/GPRS IEEE 802.11b/g Bluetooth USB
Processamento	Processador ARM11 330 MHz	Processador ARM9 134 MHz
Bateria	2.7 a 4 horas em conversação e 215 horas em <i>stand-by</i>	4 a 10 horas em conversação e 350 a 400 horas em <i>stand-by</i>
Emulador	Série 60 (S60)	JP-8.4.0 (JP8)

As três aplicações são testadas da seguinte forma: cada testador realiza os testes no emulador genérico WTK, no específico do dispositivo alvo (i.e., S60 ou JP8) e no dispositivo alvo (i.e., N95 ou G705) gerando um relatório de teste para cada ambiente. Dessa forma, cada testador gera cinco relatórios de teste para cada aplicação, totalizando trinta relatórios para análise.

5.2.2 Outras Aplicações

Para cada equipe formada na disciplina de Engenharia de Software no semestre de 2009.1, o professor atribuiu um modelo de processo de desenvolvimento (e.g., cascata, espiral, prototipação e incremental). As equipes também utilizaram em conjunto os modelos de processo, por exemplo, espiral com prototipação e incremental com a metodologia Extreme Programming (XP) [Sommerville 2007].

Para o teste das aplicações móveis, as equipes receberam um modelo de relatório de teste que contém os requisitos para serem seguidos durante a execução dos casos de teste da parte cliente da aplicação (dispositivo móvel) e foram orientadas a gerar um relatório de teste para cada ambiente de teste (i.e., emulador específico e dispositivo móvel) com os erros identificados de acordo com os requisitos propostos no Capítulo 4.

Os alunos foram divididos em quatorze equipes e as aplicações móveis foram desenvolvidas em JME. Cada equipe recebeu o dispositivo alvo da sua aplicação móvel no início do desenvolvimento. Os modelos de dispositivos alvo são celulares que possuem diferenças de fabricante (i.e., Nokia, Siemens e Sony Ericsson), de sistemas operacionais (e.g., Symbian e Windows Mobile), de funcionalidades (e.g., Wi-Fi, Bluetooth, câmera e acelerômetro), de tipos de entrada (e.g., teclado numérico, QWERT e tela sensível ao toque), de tamanho da tela (e.g., 1.9" e 3") e de resolução de tela (e.g., 176 x 220 pixels, 240 x 320 pixels e 800 x 480 pixels).

Das quatorze equipes, dez entregaram os relatórios de teste. A Tabela 5.2 resume as aplicações móveis desenvolvidas que entregaram os relatórios de teste, o modelo de processo utilizado pelas equipes, os modelos dos dispositivos alvos (i.e., celulares) das aplicações e o ambiente de teste (i.e., emulador e/ou dispositivo).

Tabela 5.2 – Características do projeto das aplicações móveis testadas

Aplicações Móveis	Modelo de Processo	Modelo do Dispositivo	Ambiente de Teste
Sistema de Gerenciamento de Pedidos de Restaurante	Cascata	Sony Ericsson Z555	Sony JP7_176x220
Controle de matrícula de estudantes universitários	Cascata	Sony Ericsson Z555	Emulador do Netbeans e celular
Sistemas de ingressos	Incremental	Sony Ericsson Xperia	Emulador S60 Dispositivo
Controle Bancário	Incremental	Sony Ericsson F305	Emulador e celular
Biólogo Móvel	Prototipação	Sony Ericsson Z780i	Emulador do Netbeans e JP8
Biblioteca Ambulante	Prototipação	Sony Ericsson 850i	Emulador
Sistema de Cadastro de Boletins de Ocorrência	Espiral com prototipação	Sony Ericsson Z780i	Emulador do Netbeans
Mobile Hotel	Espiral	Siemens CX75	Emulador do Netbeans
IMBTalk – Instant Messenger Bluetooth	Espiral	Sony Ericsson C905	Emulador JP8 e celular
Fotos Na Web	XP com Incremental	Nokia N95	Celular

5.3 Resultados da utilização dos requisitos propostos

Os Relatórios de Teste gerados são analisados com o intuito de validar como os requisitos propostos nesta dissertação ajudaram na descoberta de erros de funcionalidade e de usabilidade das aplicações móveis testadas.

Os resultados obtidos com a análise nos Relatórios de Teste gerados nas três aplicações (i.e., Shopping Móvel, Museu Móvel e Teatro Móvel) e nas dez aplicações móveis da disciplina de Engenharia de Software em 2009 são apresentados nas próximas subseções.

5.3.1 Aplicação Shopping Móvel

Requisito 1:

A aplicação testada foi desenvolvida utilizando o Modelo Cascata, portanto, não foi possível analisar o Requisito 1 que afirma que o modelo de processo de desenvolvimento deve ser focado no processo de teste. Uma vez que o processo cascata propõe um fluxo sequencial das fases do ciclo de desenvolvimento do software em que uma fase só deve iniciar quando a anterior tiver sido concluída. Dessa forma, com o Modelo Cascata, os testes só são realizados após a fase

de implementação do código.

Requisito 2:

Ao testar o Shopping Móvel no emulador genérico WTK, nos emuladores específicos S60 e JP8, e nos dispositivos N95 e G705, diferenças foram identificadas nos resultados obtidos. Os emuladores específicos S60 e JP8 simularam um ambiente mais próximo dos dispositivos móveis. Conforme a Tabela 5.3, no emulador genérico WTK foram encontrados onze erros. No emulador S60 foram identificados quinze erros e no dispositivo móvel N95, além dos quinze erros encontrados no emulador S60 foram encontrados mais três erros. Ao testar a aplicação no emulador JP8, constatamos os mesmos quinze erros encontrados no emulador S60. Entretanto, além desses quinze erros, constatamos mais quatro erros ocorridos apenas no dispositivo G705.

Tabela 5.3 – Quantidade de erros encontrados em cada ambiente na aplicação Shopping Móvel

Aplicações	WTK	S60	N95	JP8	G705
Shopping Móvel	11 erros	15 erros	18 erros	15 erros	19 erros

Além disso, foram constatadas diferenças na visualização da aplicação no WTK, S60, JP8 N95 e JP8 e são discutidas mais adiante no Requisito 9 desta subseção.

Com base nesses resultados, percebe-se que o Shopping Móvel apresentou um comportamento diferente em cada ambiente de teste (i.e., WTK, S60, N95, JP8 e G705). Os testes realizados no emulador genérico, WTK, identificaram menos erros do que quando a aplicação foi testada no seu emulador específico, WTK e JP8. Isso ocorre porque o emulador genérico não simula um ambiente próximo ao dispositivo alvo da aplicação. Por esse motivo, mais erros foram encontrados nos emuladores específicos, S60 e JP8, do que no emulador genérico, WTK.

Entretanto, somente nos testes realizados nos dispositivos alvos, uma maior quantidade de erros foi encontrada. Além disso, para cada dispositivo alvo, N95 e G705, a aplicação apresentou uma quantidade de erros diferentes.

Requisitos 3 e 4:

Para o teste do Shopping Móvel foram gerados dez relatórios, cinco para cada testador e um para cada ambiente. Os Relatório de Teste e os erros reportados no mesmo foi em conformidade com os Requisitos 3 e 4. Dessa forma, o uso das informações contidas nos requisitos propostos (e.g., versão do emulador e/ou dispositivo, frequência do erro e passo a passo para reproduzir o erro) permitiu a identificação de erros que ocorriam somente em um

determinado emulador ou dispositivo.

Requisito 5:

O consumo de **memória** foi medido nos monitores de memória dos emuladores WTK, S60 e JP8. Entretanto, não foi possível nos celulares N95 e G705, uma vez que a ferramenta Device Explorer [Sony Ericsson developer 2009] permite o monitoramento de memória no próprio dispositivo G705 via cabo conectado no computador, a medição não foi possível porque o G705 desconectava da rede sem fio (i.e., Wi-Fi) quando o cabo era conectado no mesmo. Dessa forma, como os testes do Shopping Móvel são realizados via Wi-Fi não foi possível utilizar a ferramenta Device Explorer. O consumo de memória pela aplicação só foi medido pelos emuladores (i.e., WTK, S60 e JP8) e a quantidade máxima de memória consumida foi de 11,4 MB visualizada no monitor de memória do emulador S60.

Em virtude disso, não foram identificados problemas relacionados com o consumo de memória durante os testes da aplicação já que os dispositivos móveis N95 e JP8 possuem respectivamente memória RAM de 64 MB e 128 MB.

O **processamento** da aplicação foi verificado através do tempo de resposta da aplicação para executar uma tarefa. O desempenho da aplicação foi mais lento nos dispositivos móveis, quando comparado com os emuladores, principalmente para exibir imagens de tamanho grande (e.g., 10 MB) e com altas resoluções (e.g., 12805*4724 pixels).

Quanto à visualização da aplicação na **tela**, as imagens da aplicação foram visualizadas sem cortes em todos os ambientes de teste (i.e., WTK, S60, N95, JP8 e G705). Isso só foi possível porque as imagens de altas resoluções (e.g., 1024 x 1024 pixels), armazenadas no servidor do Shopping Móvel, eram redimensionadas com uma resolução menor antes de serem enviadas para o celular. Dessa forma, as imagens eram exibidas sem cortes e sem comprometer o desempenho da aplicação, o consumo de memória e a visualização na tela do dispositivo. A Figura 5.1 ilustra uma imagem com resolução de 800*600 pixels visualizada sem cortes no Shopping Móvel.

No entanto, alguns erros de visualização do conteúdo e estrutura do menu da aplicação foram identificados nos celulares alvos e serão discutidos mais adiante nos Requisitos 13 e 16 desta subseção.

Em relação à **capacidade de armazenamento**, os testes foram feitos nos dispositivos utilizando a sua capacidade máxima de memória interna de armazenamento. A aplicação se comportou bem, pois não precisava salvar nenhum dado no dispositivo.



Figura 5.1 – Imagem com resolução original 800*600 pixels visualizada no N95

A energia consumida pela **bateria** só pode ser medida nos dispositivos móveis. Durante quatro horas de uso da aplicação com seus recursos ativados (i.e., Bluetooth, Wi-Fi, status de bateria fraca e aplicativos em espera no N95), a quantidade de energia consumida para os dois dispositivos (i.e., N95 e G705) foi, em média, dois terços do total da bateria de cada dispositivo.

A **mobilidade** da aplicação foi testada apenas dentro do laboratório utilizando o Wi-Fi com o usuário sentado e/ou em movimento sofrendo interrupções (i.e., recebimento de mensagens, chamadas recebidas, recebimento de dados via bluetooth e alarme) que foram simuladas no S60 e nos dispositivos móveis N95 e G705. No entanto, como não foi realizado o teste em campo, as interrupções mais críticas que acontecem no “mundo real” decorrentes do usuário em movimento não foram simuladas. Dessa forma, aspectos de usabilidade do ambiente móvel (e.g., iluminação, barulho e movimento físico), de desempenho na comunicação com a rede no mundo real (e.g., GPRS e 3G) que poderiam falhar se testados em campo, não foram tratados e, portanto, erros podem não ter sido identificados.

A Tabela 5.4 resume os resultados obtidos nos testes de acordo com as limitações testadas anteriormente.

Requisito 6:

Os aplicativos dos dispositivos N95 e G705 (e.g., mensagem recebida – SMS, chamada recebida, alarme, *Bluetooth*, dentre outros aplicativos do dispositivo) continuaram funcionando após a instalação e o uso da aplicação Shopping Móvel.

Tabela 5.4 – Resultados dos testes de acordo com as limitações testadas no Shopping Móvel

Limitação	Resultados do Shopping Móvel
Memória	Máximo: 11,4 MB no S60
Processamento	Mais lento no dispositivo móvel
Tela	Imagens foram visualizadas sem cortes ou distorções Erros só visualizados no dispositivo móvel
Bateria	Média 2/3 do total em 4 horas de uso
Capacidade de Armazenamento	Satisfatório
Largura de Banda	Wi-Fi no dispositivo móvel
Mobilidade	WTK e JP8: Não aplicável S60, N95 e G705: apenas no laboratório

Requisito 7:

No teste de laboratório realizado no emulador puderam ser identificados erros de funcionalidades da aplicação e algumas questões de usabilidade (e.g., não exibição de mensagens de erro e informações repetidas). Além disso, no teste de laboratório feito nos dispositivos móveis, mais erros inerentes ao contexto móvel puderam ser identificados em relação às limitações dos dispositivos, tais como o tempo de resposta da aplicação, consumo de memória, consumo de bateria e interrupções.

Requisito 8:

O Shopping Móvel foi desenvolvido utilizando o Modelo Cascata na disciplina de ES em 2006 por alunos já experientes no desenvolvimento de aplicações móveis. Em virtude disso, percebe-se uma maior preocupação em tratar aspectos inerentes do ambiente móvel como, por exemplo, o redimensionamento de imagens conforme apresentado no Requisito 2 desta subseção. No entanto, eles não incluíram o teste de usabilidade no ciclo de desenvolvimento (Requisito 8) e erros triviais de usabilidade, apresentados mais adiante no Requisito 9 desta subseção, foram reportados pelos testadores durante os testes do Shopping Móvel e, com isso, acredita-se que a não conformidade com o Requisito 8 foi um fator que contribuiu para que muitos erros de usabilidade fossem identificados.

Requisito 9:

Os erros encontrados foram em relação à visualização e falta de consistência das mensagens na aplicação que impedem o usuário de saber o que fazer em determinada situação na aplicação. Esses erros são descritos a seguir:

Dificuldade de visualização das mensagens: a aplicação Shopping Móvel exibe suas mensagens de erro no início da tela, deslocando o seu conteúdo para baixo. Quando os testes são realizados nos emuladores específicos S60 e JP8 ou nos dispositivos móveis N95 e G705, não é possível visualizar as mensagens de erro devido ao tamanho da tela. No entanto, quando a aplicação é testada no emulador genérico WTK, todo conteúdo da página, incluindo as mensagens de erro, são visualizadas completamente, pois no emulador genérico WTK, o tamanho da tela é maior e, portanto, exibe uma maior quantidade de informação do que a tela do dispositivo móvel.

Conforme ilustra a Figura 5.2, quando o usuário se cadastra no Shopping Móvel e preenche, por exemplo, a senha de forma incorreta, a mensagem de erro é exibida no topo da tela e a mesma só é visualizada completamente no WTK.



Figura 5.2 – Shopping Móvel: Visualização de mensagem de erro no WTK

No JP8 e G705 não é possível visualizar a mesma mensagem de erro exibida no WTK conforme pode ser observado na Figura 5.3, além disso, no G705, o usuário só nota que houve uma resposta da aplicação se perceber que a barra de rolagem desceu e, nesse caso, existe alguma informação no início da tela.

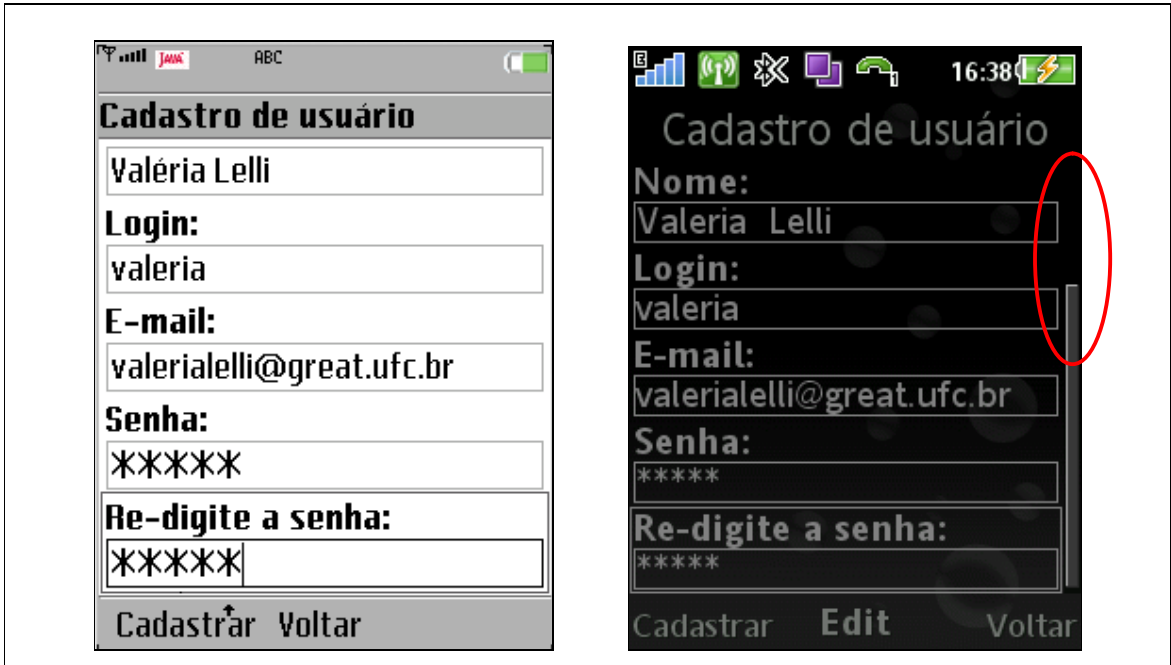


Figura 5.3 - Visualização de mensagem de erro no JP8 e no G705

Já no N95, é possível visualizar apenas uma parte dessa mensagem, diferentemente do que aconteceu no emulador específico, S60, conforme ilustrado na Figura 5.4. Neste cenário, o S60 simulou um comportamento atípico ao exibir menos informações na tela do que o seu dispositivo alvo, o N95.

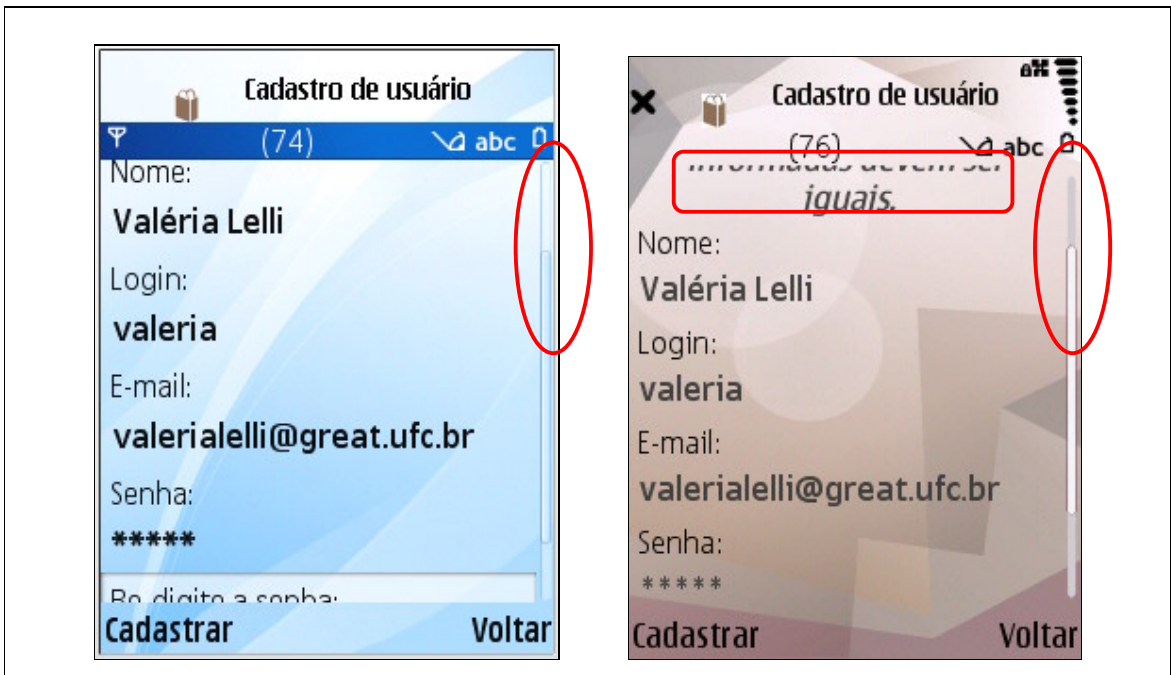


Figura 5.4 – Visualização de mensagem de erro no S60 e no N95

Mensagens inconsistentes: quando o usuário efetua uma compra na qual excede a quantidade de estoque do produto aparece uma mensagem para o usuário informando a quantidade excedida. Após isso, toda vez que o usuário for comprar outro produto essa mesma mensagem continuará na tela.

Requisito 10:

Nos emuladores WTK, S60 e JP8 foi possível testar algumas interrupções (e.g., *pause* – WTK, JP8 e S60, status da bateria e alarme – S60). No entanto, os emuladores apresentaram comportamentos divergentes, no emulador genérico WTK e no emulador JP8, a aplicação sempre voltava para a tela inicial após a interrupção. Já nos dispositivos N95 e G705 e no emulador S60, a aplicação manteve o seu estado anterior a interrupção.

Embora os emuladores WTK e JP8 sejam de empresas diferentes, eles possuem aspectos semelhantes, como por exemplo, o JP8 utiliza o mesmo monitor de memória do WTK. Dessa forma, acredita-se que o problema da aplicação não manter o seu estado ao sofrer interrupções nos mesmos é devido às configurações no ambiente dos emuladores.

Requisito 11:

O Shopping Móvel não suporta os dois formatos (i.e., retrato e paisagem). No entanto, os dispositivos N95 e G705 possuem a função auto-rotação que permite que as aplicações sejam visualizadas nesses formatos.

Durante os testes, a aplicação foi visualizada nos dois formatos apenas no emulador S60 e no dispositivo N95 e não foram identificados erros diferentes dos já encontrados no modo retrato. Já no G705 não foi possível visualizar o Shopping Móvel no modo paisagem. Acredita-se que o suporte deste dispositivo para as duas orientações é somente para aplicações desenvolvidas em Flash Lite ou que utilizam a função sensor de movimento.

Em virtude disso, percebe-se que determinados modelos de dispositivos já permitem visualizar as aplicações nos dois formatos. Entretanto, essa funcionalidade depende do modelo de dispositivo e ainda assim depende da tecnologia utilizada pela aplicação. Dessa forma, mesmo que o dispositivo alvo da aplicação forneça o suporte para auto-rotação, ela pode não ser visualizada corretamente nos dois formatos.

Requisito 12:

Tanto nos emuladores quanto nos dispositivos, o Shopping Móvel pede inicialmente para estabelecer comunicação com a rede. Porém, essas solicitações de comunicação dependem de como as políticas de segurança do ambiente (e.g., toolkit ou KVM – Kilo Virtual Machine)

exigidas ou não pela aplicação foram configuradas pelo desenvolvedor ou pelo fabricante do dispositivo móvel.

Requisito 13:

No emulador genérico WTK, o conteúdo foi visualizado sem o uso da barra de rolagem e com um bom contraste em relação ao plano de fundo. Já nos demais emuladores e dispositivos, a aplicação é visualizada com o uso da barra de rolagem e os erros em razão desse uso foram identificados conforme apresentado no Requisito 9 desta subseção.

No entanto, no dispositivo N95 ocorreu um erro constatado apenas nesse ambiente que prejudicou a visualização do texto da aplicação. O título que aparece no topo de cada tela é cortado e quebrado em várias linhas, impossibilitando a leitura das informações. Conforme observado na Figura 5.5, no S60 o subtítulo “Promoções” foi visualizado normalmente, porém no N95, esse mesmo título apareceu quebrado em várias linhas.



Figura 5.5 – Diferenças de visualização de texto no S60 e no N95

No entanto, no dispositivo N95 ocorreu um erro constatado apenas nesse ambiente que prejudicou a visualização do texto da aplicação. O título que aparece no topo de cada tela é cortado e quebrado em várias linhas, impossibilitando a leitura das informações. Conforme observado na Figura 5.5, no S60 o subtítulo “Promoções” foi visualizado normalmente, porém no N95, esse mesmo título apareceu quebrado em várias linhas.

Dessa forma, a exibição do texto dificultou a sua leitura e também forçou o usuário a rolar um texto quebrado em várias linhas. Portanto, em ambas as situações a usabilidade da aplicação foi prejudicada. Além disso, o Shopping Móvel no N95 apresentou um comportamento diferente do emulador específico S60, o que reforça a necessidade do uso do Requisito 2 de testar a aplicação tanto no emulador quanto no dispositivo móvel.

Já no dispositivo G705, ocorreram dois erros exclusivos desse ambiente. O primeiro é que alguns nomes de lojas e/ou produtos apareceram alinhados à direita e com isso eles passaram despercebidos durante a aplicação. Na Figura 5.6, o nome da loja “Marítima” aparece alinhado à direita, dessa forma, o usuário pode não perceber a existência dessa loja.



Figura 5.6 – Nome da loja alinhado à direita no G705

O segundo erro é exemplificado na Figura 5.7, quando o usuário visualiza algum item da aplicação pela primeira vez, ele clica no botão “Selecionar” da *softkey* do meio. Se ele visualizar o que deseja e voltar para a tela anterior, além desta opção “Selecionar” haverá outra opção “Selecionar” na *softkey* da esquerda. Caso, o usuário deseje visualizar a tela novamente, ele deverá clicar nessa nova opção “Selecionar”, pois a opção do meio não funciona mais. À medida que o usuário clica em uma opção “Selecionar”, visualiza o conteúdo desejado e volta para a tela anterior, a aplicação adiciona uma opção “Selecionar” na *softkey* da esquerda e a mesma a partir de agora valerá para acessar aquele conteúdo. No entanto, a lista de opções “Selecionar” criada na *softkey* da esquerda é atualizada apenas quando se clica no segundo item exibido na tela. Caso o usuário deseje visualizar o primeiro item, ele tem que clicar no item abaixo para atualizar a lista.

Devido a esse erro não é possível visualizar produtos ou lojas que possuam apenas um item.

Além disso, o plano de fundo da aplicação visualizado nos dispositivos foi de acordo com o perfil escolhido pelo usuário do próprio dispositivo e, nesse caso, se a aplicação não definir um plano durante o desenvolvimento, um bom contraste dependerá do perfil.

Através do uso deste Requisito foi possível encontrar erros que afetaram a usabilidade do Shopping Móvel e que só foram identificados em determinados ambientes.

Requisito 14:

Quando o cliente efetua a compra de um produto no Shopping Móvel não são confirmados os dados da compra, tais como endereço de entrega, quantidade do produto e valor.

Dessa forma, muitos erros foram ocasionados pela falta de confirmação dessas informações.

Requisito 15:

A aplicação testada não possui som. No entanto, no dispositivo N95 quando o usuário digita um valor não permitido é emitido um som de alerta, e o mesmo acontece com o quando o usuário efetua uma compra no dispositivo G705. Isso ocorreu devido às configurações do próprio dispositivo que dependendo do tipo de ação habilita sons de alerta para o usuário.

Requisito 16:

Os rótulos das teclas *softkeys* da aplicação se adaptaram conforme o dispositivo móvel. No entanto, o G705 possui a tecla *softkey* do meio além das *softkeys* da direita e da esquerda. Dessa forma, no dispositivo G705 a estrutura do menu da aplicação ficou confusa com duas opções de seleção tanto na *softkey* da esquerda quanto na *softkey* do meio conforme apresentado no Requisito 13. Além disso, nos dois modelos de dispositivos testados, as ações positivas e as ações negativas foram utilizadas conforme descritas no Requisito 16 da Seção 4.2.2.

Embora a aplicação tenha usado as *softkeys* conforme o dispositivo e que mesmo assim erros relacionados com a estrutura do menu acessado pelas *softkeys* foram identificados e algumas vezes, impossibilitando de utilizar a aplicação.

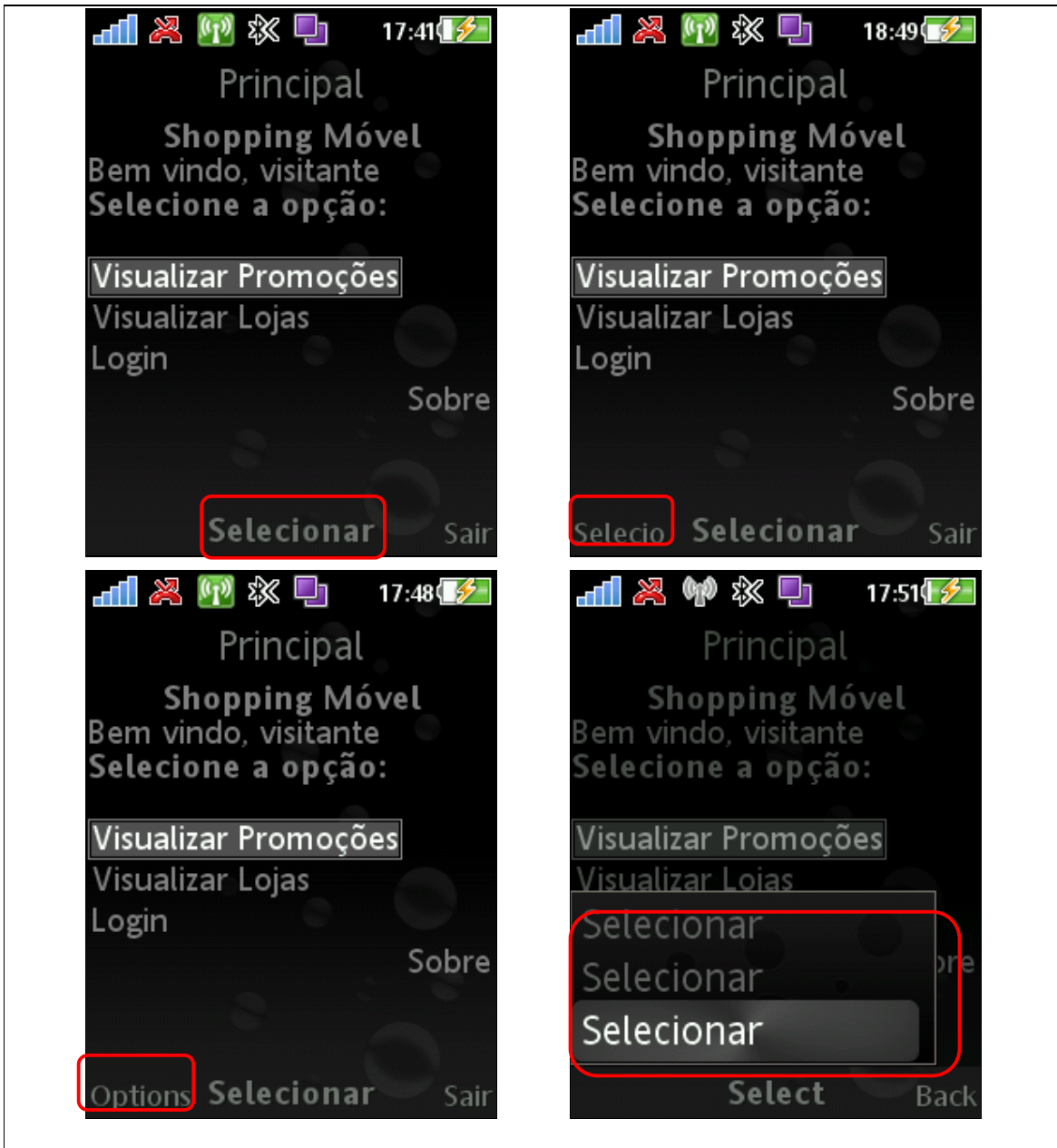


Figura 5.7 – Menu de opções “Selecionar” na *softkey* da esquerda no G705

5.3.2 Aplicação Museu Móvel

Requisito 1: resultado semelhante ao Requisito 1 da aplicação Shopping Móvel.

Requisito 2:

Ao testar ao Museu Móvel no emulador genérico WTK, nos emuladores específicos S60 e JP8, e nos dispositivos N95 e G705, diferenças foram identificadas nos resultados obtidos. Conforme a Tabela 5.5, no emulador genérico WTK foram encontrados treze erros. No

emulador S60 e N95 foram identificados os mesmos erros e mais um erro de usabilidade, totalizando quatorze erros. No JP8 foram encontrados onze erros já ocorridos no WTK, no G705, além desses onze erros foram identificados mais três erros exclusivos desse ambiente.

Tabela 5.5 – Quantidade de erros encontrados em cada ambiente do Museu Móvel

Aplicações	WTK	S60	N95	JP8	G705
Museu Móvel	13 erros	14 erros	14 erros	11 erros	14 erros

Com base nesses resultados, percebe-se que podem ocorrer situações em que a quantidade de erros encontrada seja a mesma no emulador específico e no dispositivo móvel. Entretanto, a aplicação pode apresentar comportamentos diferentes de acordo com o modelo de dispositivo alvo e, portanto, os erros identificados podem ser distintos.

Requisitos 3 e 4: resultado semelhante aos Requisitos 3 e 4 da aplicação Shopping Móvel.

Requisito 5:

O Museu Móvel suporta três formatos de imagem: JPG, PNG e BMP, mas as imagens não eram redimensionadas no servidor como era feito no Shopping Móvel. Isso impactou bastante no consumo de memória pela aplicação ao abrir imagens “pesadas” e de altas resoluções. Em virtude disso, a memória máxima dos emuladores foi configurada para 32 MB com intuito de tornar o ambiente de teste mais próximo do dispositivo alvo.

O consumo de memória foi medido de forma semelhante à aplicação Shopping Móvel. A quantidade máxima de **memória** consumida pela aplicação foi de 31,5 MB visualizada no monitor de memória do emulador S60. Tanto nos emuladores quanto nos celulares as imagens com tamanho superior a 1MB (geralmente do tipo BMP) e de altas resoluções (e.g., 1280*1024) não foram visualizadas no Museu Móvel. No celular G705 ao tentar abrir essas imagens, a aplicação fechava após exibir a seguinte mensagem de erro: “The applications uses too much memory”. No N95 o comportamento foi semelhante, mas a aplicação não exibia mensagem de erro ao fechar. Nas situações apresentadas, a memória consumida pela aplicação usou o limite máximo de memória RAM dos celulares (memória RAM de 64 MB no N95 e 128 MB no G705) e, portanto, não pode mais ser executada e fechou.

Os testes no N95 também foram feitos com a aplicação instalada no cartão de memória e ela se comportou do mesmo modo quando instalada na memória do telefone.

O **processamento** do Museu Móvel foi verificado através do tempo de resposta da

aplicação para executar uma tarefa. O desempenho da aplicação foi mais lento S60, N95, JP8, G705 quando comparado com o WTK. No WTK, o tempo de resposta para abrir as imagens “pesadas” e de altas resoluções foi, em média, de 10 segundos, já nos demais emuladores e celulares elas tiveram um tempo de resposta, em média, superior a 15 segundos.

Além disso, outro erro identificado em todos os ambientes de teste (i.e., WTK, S60, N95, JP8 e G705) foi que algumas imagens abriram com atraso. Por exemplo, ao visualizar uma obra e a mesma não abrir, ao sair da tela e executar outras tarefas, essa obra que não abriu é exibida na tela atual. Essa situação é ilustrada na Figura 5.8, enquanto o usuário fazia uma busca por autor, a obra que não havia sido visualizada anteriormente abriu e se sobrepôs a tela em uso, prejudicando a legibilidade do texto na aplicação.

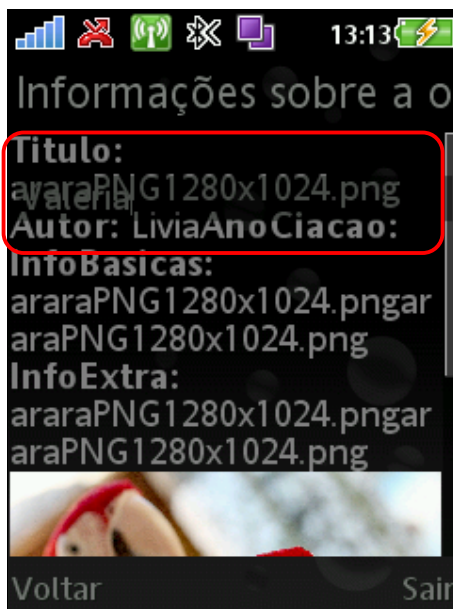


Figura 5.8 – Telas sobrepostas do Museu Móvel no G705

Quanto à visualização da aplicação na **tela**, percebe-se que uma menor quantidade de texto é visualizada na tela nos dispositivos alvos, N95 e G705, quando comparados com seus emuladores específicos, S60 e JP8, respectivamente.

Além disso, em cada dispositivo alvo (i.e., N95 e G705) a forma como as informações são apresentadas na tela foi diferente. Por exemplo, nas Figuras 5.9 e 5.10, o título da tela “Informações sobre a obra” exibido no N95 apresentou tamanho de fonte diferente do G705. Outra diferença observada ilustrada na Figura 5.9 foi que embora esse título não fosse visualizado com cortes no JP8, ele apareceu cortado na tela do seu dispositivo alvo, G705.

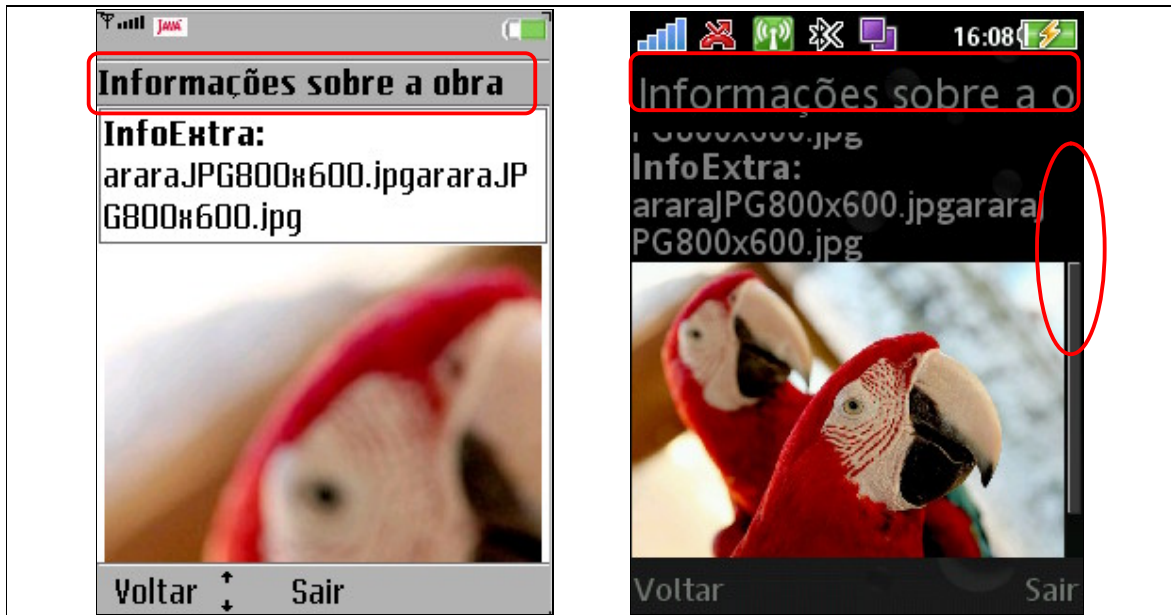


Figura 5.9 – Imagens visualizadas no JP8 e no G705

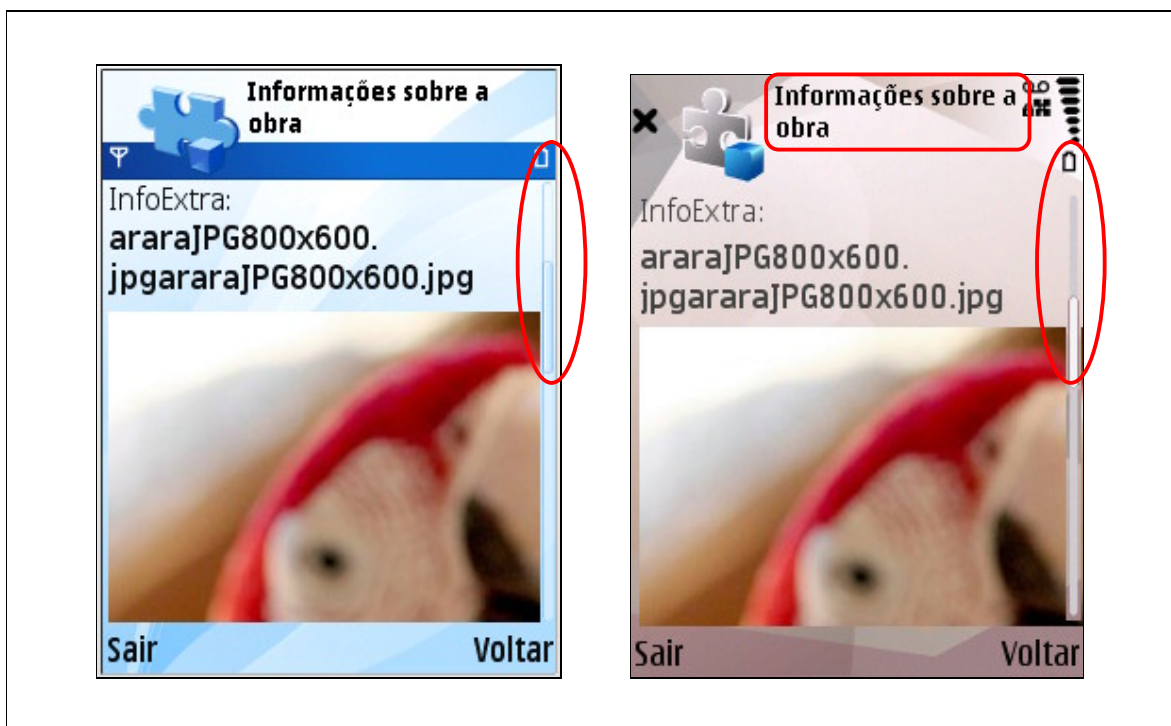


Figura 5.10 – Imagens visualizadas no S60 e no N95

Como mencionado anteriormente, as imagens do Museu Móvel não eram redimensionadas para serem visualizadas em uma tela menor. Dessa forma, as imagens com resoluções acima de 240*320 pixels são visualizadas com cortes do lado direito e as imagens com altas resoluções (e.g., 800*600 pixels), além de cortadas são distorcidas, pois a aplicação não dá

suporte para redimensionamento de imagens e nem suporte para barra de rolagem vertical, conforme observado nas Figuras 5.9 e 5.10. Já o G705 apresentou um comportamento diferente do seu emulador específico, as imagens que aparecem cortadas e distorcidas no JP8 foram visualizadas nele sem cortes ou distorções conforme observado na Figura 5.9.

Em relação à **capacidade de armazenamento**, os testes foram feitos nos dispositivos utilizando a sua capacidade máxima de memória interna de armazenamento. O erro identificado ao abrir imagens “pesadas” foi devido à limitação da memória RAM dos celulares e não porque sua capacidade de armazenamento estava cheia, pois o erro persistiu ao esvaziarmos a memória de armazenamento.

Em relação à quantidade de energia, o consumo medido durante o uso da aplicação foi, em média, um terço da **bateria** consumida para os dois dispositivos em duas horas de uso da aplicação com seus recursos ativados (i.e., Bluetooth, Wi-Fi, status de bateria fraca e aplicativos em espera no N95) simulando interrupções.

Em relação à **mobilidade** do Museu Móvel, o resultado é semelhante ao Requisito 5 do Shopping Móvel.

A Tabela 5.6 resume os resultados obtidos nos testes de acordo com as limitações apresentadas acima.

Tabela 5.6 – Resultados dos testes de acordo com as limitações testadas no Museu Móvel

Limitação	Resultados do Museu Móvel
Memória	Máximo: 31,5 MB no S60. Erro de memória cheia nos dispositivos N95 e G705
Processamento	Mais lento no dispositivo móvel
Tela	Textos cortados Imagens com altas resoluções, pesadas e distorcidas
Bateria	Média 1/3 do total em 2 horas de uso
Capacidade de Armazenamento	Satisfatório
Largura de Banda	Wi-Fi no dispositivo móvel
Mobilidade	WTK e JP8: não aplicável S60, N95 e G705: apenas em laboratório

Requisito 6: resultado semelhante ao Requisito 6 da aplicação Shopping Móvel.

Requisito 7:

No teste feito em laboratório utilizando emulador e dispositivo foi possível testar as seguintes características:

- *Funcionalidades que não dependem do teste em campo:* formatos, tamanhos e resoluções de imagens que não abriam ou que foram visualizadas com distorção e cortadas.
- *Limitações:*
 - Processamento: tempo de resposta da aplicação
 - Memória: memória consumida.
 - Mobilidade: interrupções na aplicação
 - Bateria: medição do consumo
 - Tela: imagens e textos cortados
- *Usabilidade:* ausência de mensagem de erro quando uma obra não podia ser visualizada, uso inapropriado do botão “Voltar”, dentre outras.

Requisito 8:

O Museu Móvel foi desenvolvido utilizando o Modelo Cascata na disciplina de ES em 2006 por alunos que não tinham experiência no desenvolvimento de aplicações móveis. Dessa forma, não houve um tratamento da aplicação no redimensionamento das imagens com altas resoluções semelhante ao que foi feito na aplicação Shopping Móvel e, portanto, as imagens foram visualizadas com cortes e distorcidas na tela do dispositivo conforme apresentando no Requisito 2 desta subseção.

Além disso, também não foi incluído o teste de usabilidade no ciclo de desenvolvimento (Requisito 8) e, portanto, erros triviais de usabilidade foram identificados e são apresentados mais adiante nos Requisitos 9 e 13 desta subseção.

Requisito 9:

Ao testar a aplicação conforme este requisito, os seguintes erros foram identificados:

A ausência de mensagem de erro: quando o usuário não consegue visualizar uma obra, a tela da aplicação fica parada.

Uso inadequado do rótulo “Voltar”: quando o usuário digita o nome de um autor para a “busca” e visualiza as obras daquele autor e escolhe uma obra para ver os detalhes. Se ele clicar

no botão “Voltar” para visualizar as demais obras daquele autor, a aplicação não volta para as obras anteriores do autor e sim para a tela inicial de busca para que o usuário entre com os dados novamente.

Deficiência no foco de seleção: o foco de seleção não é sintonizado com o cursor da aplicação. Por exemplo, conforme observado na Figura 5.11, se o usuário visualizar as obras do Museu e percorrer a lista das obras apresentadas, o foco da aplicação acompanha, mas a seleção não. Nesse caso, para selecionar a obra o usuário tem que clicar uma vez no botão “enter” para que a opção fique selecionada e novamente no “enter” para visualizar a obra. Esse erro de usabilidade pode ser visualizado em todos os ambientes de teste.

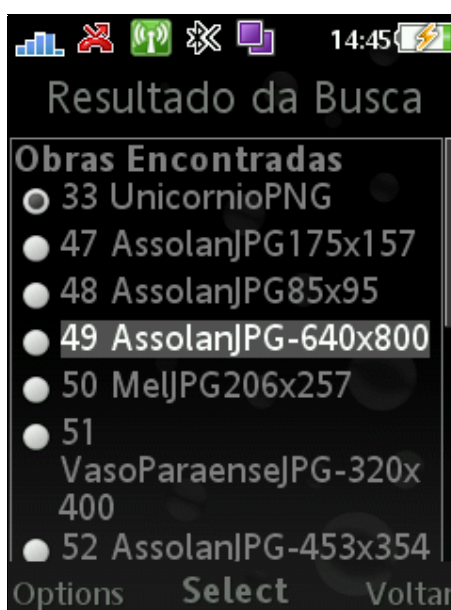


Figura 5.11 – Deficiência no foco de seleção no G705

Atraso na visualização das obras: no caso de o usuário desistir de visualizar uma obra que está demorando a carregar uma imagem e continuar usando a aplicação para visualizar outras obras de outros autores, a obra que ele desistiu de visualizar aparece na tela quando consegue carregar a imagem mesmo após ele ter saído da tela em que a imagem está carregando. Situação já ilustrada na Figura 5.8.

Em todas as situações apresentadas acima, a interação do usuário com a aplicação é prejudicada.

Requisito 10: resultado semelhante ao Requisito 10 da aplicação Shopping Móvel.

Requisito 11: resultado semelhante ao Requisito 11 da aplicação Shopping Móvel.

Requisito 12: além de apresentar resultados semelhantes ao observado no Requisito 12

da aplicação Shopping Móvel, o Museu móvel tem um acesso restrito para as informações das obras do museu e solicita uma senha inicial para o acesso que só é possível dentro do museu.

Requisito 13:

Os erros identificados são apresentados a seguir:

Visualização do texto: conforme observado na Figura 5.12, no WTK o texto foi exibido na íntegra sem cortes e nenhuma imagem do tipo BMP abriu. No S60 e no N95, os nomes das obras quando extensos aparecem cortados com um grande espaço entre o nome da obra e o “radio button” para selecioná-la. Já no JP8 e no G705 os nomes das obras aparecem quebrados em várias linhas em vez de cortados com pontinhos e não houve espaço entre o radio Button e o início da palavra conforme ilustrado na Figura 5.13. Nas Figuras 5.12 e 5.13 percebe-se que embora a visualização da aplicação na tela não seja igual no emulador específico e no celular, aquele simula uma interface bem mais próxima da plataforma alvo.

Barra de rolagem: uma menor quantidade de texto é exibida na tela nos dispositivos alvos: N95 e G705 quando comparados com seus emuladores específicos S60 e JP8, respectivamente. a aplicação é visualizada com uso de barra de rolagem apenas no S60, N95 e G705. Nesse cenário, conforme apresentado no S60 e N95 na Figura 5.10, quando o Museu exhibe a imagem de uma obra com grande resolução, o usuário além de não visualizar a imagem como um todo, ele tem que descer a barra de rolagem até poder visualizar as demais informações da obra.

Plano de fundo: Não foi definido na aplicação. Dessa forma, o plano de fundo dependerá do perfil escolhido no S60, N95, JP8 e G705.

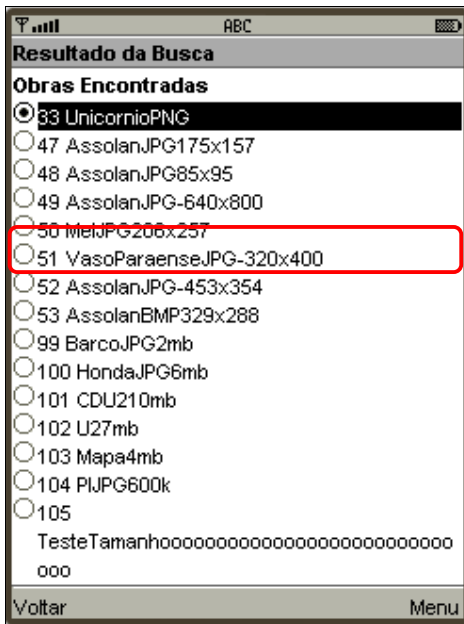
Requisito 14:

No Museu Móvel, o usuário apenas visualiza as obras do museu através de buscas, e dessa forma caso, não se aplicou testá-la conforme o Requisito 14.

Requisito 15:

A aplicação não possui som prejudicando a usabilidade da aplicação, principalmente na exibição de mensagens de alerta, aviso ou erro para o usuário móvel.

WTK



S60

N95

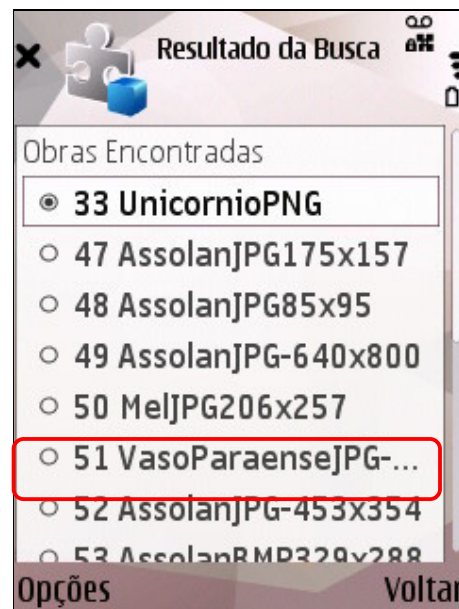
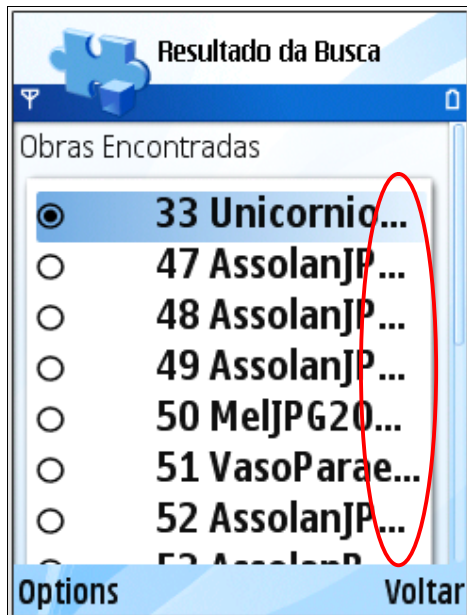


Figura 5.12 – Museu Móvel: Visualização do Texto no WTK, no S60 e no N95

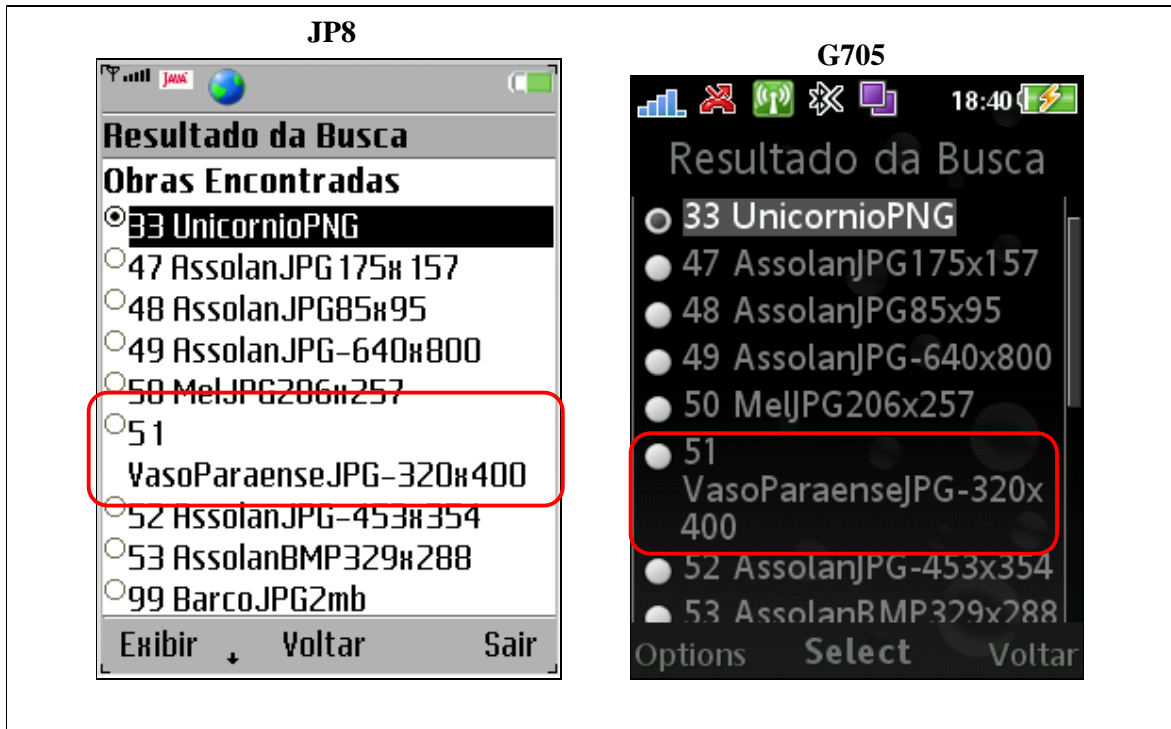


Figura 5.13 – Visualização do Texto no JP8 e no G705

Requisito 16:

Os rótulos das *softkeys* não se adaptaram conforme os dispositivos. Na estrutura do menu de acesso aos aplicativos do N95, as teclas *softkeys* da esquerda são usadas para ações positivas e as da direita para ações negativas. No entanto, no Museu Móvel as *softkeys* não seguiram o padrão desse celular. As *softkeys* da esquerda, além de possuírem ações positivas (e.g., “Buscar”, “Registrar”, “Opções” e “Escolher”) também possuem ações negativas (e.g., “Sair”). As *softkeys* da direita apenas para ações negativas (e.g., “Voltar” e “Sair”).

De modo semelhante, o Museu Móvel também não se adaptou ao padrão do G705, que além das *softkeys* da direita e esquerda também possui a tecla *softkey* do meio. Na aplicação, as teclas *softkeys* da esquerda são utilizadas para ações positivas (e.g., “Buscar”, “Registrar” e “Opções”) e negativas (e.g., Sair e Voltar) e as *softkeys* da direita apenas para a ação “Sair” e a *softkey* do meio ficou exclusivamente para as ações positivas (e.g., “Ok”, “Select” e “Edit”). Conforme ilustrado na Figura 5.14, a falta de padronização atrapalhou o usuário que está habituado a usar o G705, pois ao utilizar a aplicação no G705 quando ele desejava “voltar” na aplicação, ele pressionava a *softkey* da direita em vez de pressionar a da esquerda, dessa forma em vez de voltar para a tela anterior, o usuário acabava saindo da aplicação.

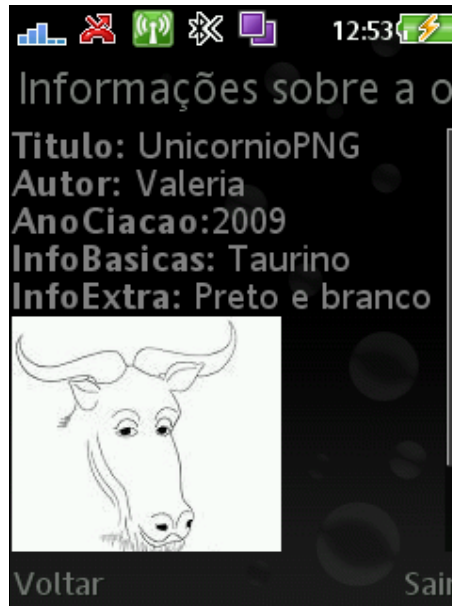


Figura 5.14 – Museu Móvel: rótulos das *softkeys* no G705

Além disso, apenas no G705 apareceram rótulos no menu, tais como “Select” e “Edit” que não pertenciam ao menu original da aplicação, além de que esses rótulos eram exibidos no idioma configurado pelo celular e nem sempre funcionavam.

5.3.3 Aplicação Teatro Móvel

Requisito 1: resultado semelhante ao Requisito 1 da aplicação Shopping Móvel e Museu Móvel.

Requisito 2:

Ao testar o Teatro Móvel nos emulador específico S60 e N95, o comportamento da aplicação foi semelhante, e dessa forma os mesmos erros foram identificados. Já no celular G705, foram encontrados erros de usabilidade que não ocorreram no emulador específico JP8. Conforme a Tabela 5.7, no emulador genérico WTK foram encontrados nove erros. No emulador S60 e N95 foram identificados 17 erros iguais, dos quais dez erros encontrados são diferentes dos erros encontrados no WTK. No JP8 foram encontrados dezesseis erros, dos quais apenas oito foram identificados no WTK. No G705, além desses dezesseis erros foram identificados mais dois erros de usabilidade exclusivos desse ambiente, totalizando dezoito erros.

Tabela 5.7 – Quantidade de erros encontrados em cada ambiente no Teatro Móvel

Aplicações	WTK	S60	N95	JP8	G705
Teatro Móvel	9 erros	17 erros	17 erros	16 erros	18 erros

Os resultados apresentados foram semelhantes ao observado no Requisito 2 da aplicação Museu Móvel.

Requisitos 3 e 4: resultado semelhante aos Requisitos 3 e 4 da aplicação Shopping Móvel e Museu Móvel.

Requisito 5:

O Teatro Móvel diferente das aplicações Shopping Móvel e Museu Móvel não possui imagens, apenas informações textuais. O consumo de **memória** foi medido de forma semelhante à aplicação Shopping Móvel. A quantidade máxima de *memória* consumida pela aplicação foi aproximadamente 10MB visualizada no monitor de memória do emulador específico S60. Dessa forma, não foram encontrados erros referentes a “estouro de memória” durante a execução dos testes do Teatro Móvel conforme identificado no Museu Móvel.

O **processamento** do Teatro Móvel foi verificado através do tempo de resposta da aplicação para executar uma tarefa. A aplicação teve um bom tempo de resposta para as requisições nos ambientes WTK, S60, N95, JP8 e G705. Acredita-se que devido à natureza da aplicação que contém apenas informações textuais, o seu desempenho foi bem satisfatório quando comparado ao Shopping Móvel e Museu Móvel.

A quantidade de texto visualizada na **tela** do WTK é bem maior (aproximadamente o dobro) do que a quantidade exibida nos emuladores específicos e dispositivos alvos. No N95 e no seu emulador específico S60, assim com no G705 e no seu emulador específico JP8, a aplicação foi exibida de forma semelhante, simulando uma visualização bem próxima dos seus dispositivos alvo. Por exemplo, o Teatro Móvel apresentou texto cortado e com três pontinhos no S60 e no N95, além de grandes espaços na tela antes da descrição das peças de teatro.

Os testes foram realizados nos dispositivos N95 e G705 utilizando a sua capacidade máxima de memória interna de armazenamento. Durante os testes não foram encontrados erros referentes à **capacidade de armazenamento** do dispositivo, pois a aplicação não precisava salvar dados no dispositivo móvel. No entanto, o N95 exibia mensagens de alertas para informar que a memória interna do dispositivo estava cheia.

Os testes para medir o consumo da **bateria** do dispositivo móvel foram feitos com seus recursos ativados (i.e., Bluetooth, Wi-Fi, status de bateria fraca e aplicativos em espera no N95). A energia consumida pela aplicação foi, em média, um terço da bateria em duas horas de uso no N95. Já o G705 consumiu, em média, metade do tempo de vida da bateria em duas horas de uso.

Em relação à **mobilidade** do Teatro Móvel, o resultado é semelhante ao Requisito 5 do

Shopping Móvel e Museu Móvel.

A Tabela 5.8 resume os resultados obtidos nos testes de acordo com as limitações apresentadas acima.

Tabela 5.8– Resultados dos testes de acordo com as limitações testadas do Teatro Móvel

Limitação	Resultados do Museu Móvel
Memória	Máximo: 10 MB no S60.
Processamento	Resultado semelhante no WTK, S60, N95, JP8 e G705
Tela	Textos cortados com três pontinhos Grandes espaços em branco no texto
Bateria	Média ^{1/3} do total em 2 horas de uso no N95 Média ^{1/2} do total em 2 horas de uso no G705
Capacidade de Armazenamento	Satisfatório. Com alertas de mensagem do dispositivo informando que a capacidade estava cheia
Largura de Banda	Wi-Fi no dispositivo móvel
Mobilidade	WTK e JP8: não aplicável S60, N95 e G705: apenas em laboratório

Requisito 6: resultado semelhante ao Requisito 6 da aplicação Shopping Móvel.

Requisito 7:

No teste realizado em laboratório utilizando o emulador no computador e o dispositivo móvel, foi possível testar as seguintes características:

- *Limitações:* resultado semelhante ao Requisito 7 do Museu Móvel.
- *Usabilidade:*
 - Texto cortado com espaços grandes entre as linhas.]
 - Ausência de mensagens de erros e mensagens de informação inadequadas.
 - Falta dos rótulos “voltar” nas telas da aplicação, além de rótulos na estrutura do menu que não funcionam.

Requisito 8:

O Teatro Móvel foi desenvolvido utilizando o Modelo Cascata na disciplina de ES em 2006 por alunos que não tinham experiência no desenvolvimento de aplicações móveis. O teste de usabilidade não foi incluído no ciclo de desenvolvimento (Requisito 8) da mesma forma que

no Shopping Móvel e no Museu Móvel e, portanto, erros triviais de usabilidade foram identificados e são apresentados mais adiante nos Requisitos 9 e 13 desta subseção.

Requisito 9:

Ao testar a aplicação observando este requisito no WTK, S60, N95, JP8 e G705, foram identificados os seguintes erros: falta de objetividade nas mensagens, uso inadequado do rótulo “voltar” e rótulos inconsistentes das *softkeys*. Por exemplo, quando o testador preencheu algum campo do cadastro com dados inválidos a aplicação exibiu a seguinte mensagem: “Atenção! Erro, tente novamente!” informando ao testador que ocorreu um erro conforme ilustrado na Figura 5.15. No entanto, ele não sabe qual dos campos do cadastro que digitou os dados inválidos. Além disso, na tela da mensagem de erro aparece o rótulo “OK” para a *softkey* da esquerda e na da direita, o rótulo “Voltar”, se o usuário pressionar a tecla da esquerda ou a da direita, a aplicação volta para a tela que contém a lista das peças de teatro em vez de voltar para a tela do cadastro conforme observado na Figura 5.16.

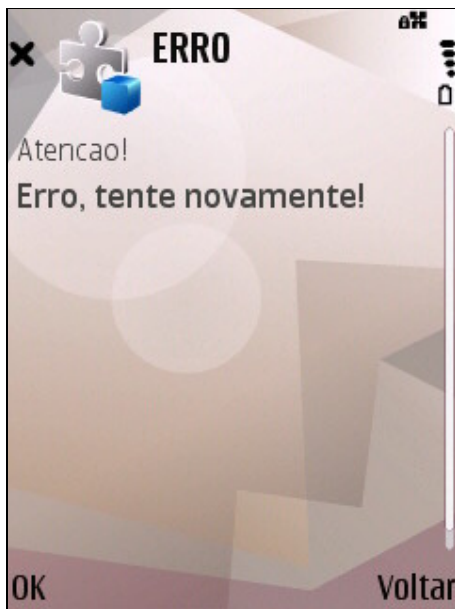


Figura 5.15 – Mensagem de erro

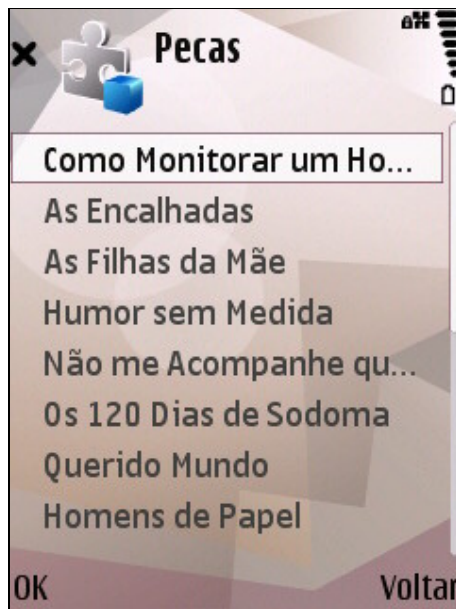


Figura 5.16 – Lista das peças

Outro erro identificado foi a ausência de mensagens de erros, por exemplo, quando o usuário consultou uma reserva e não havia nenhuma para ele, a aplicação não exibiu mensagem informando que não existia reserva, a tela ficou em branco conforme ilustrado 5.17. Além disso, a tela inicial da aplicação não possuía um rótulo para o usuário sair da aplicação como pode ser observado na Figura 5.18. A função “sair” somente é encontrada na tela de cadastro do Teatro Móvel.

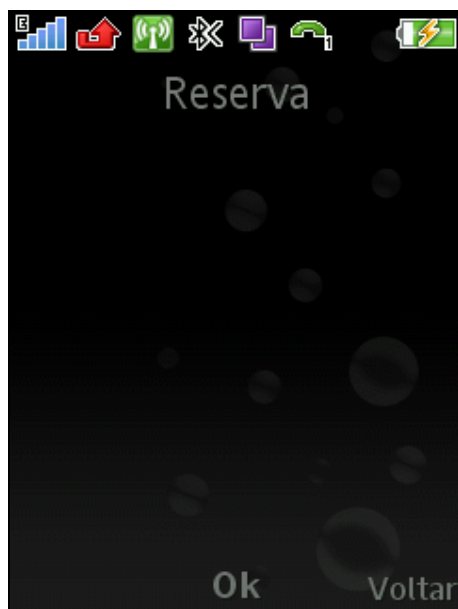


Figura 5.17 – Tela de reservas

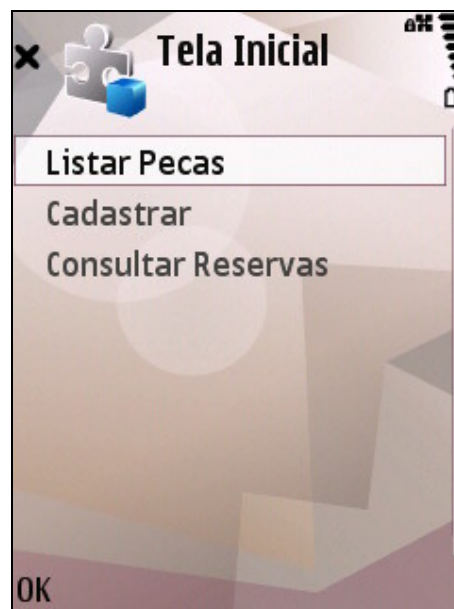


Figura 5.18 – Tela inicial do Teatro Móvel

Requisito 10: resultado semelhante ao Requisito 10 da aplicação Shopping Móvel e Museu Móvel.

Requisito 11:

O Teatro Móvel não suporta os dois formatos (i.e., retrato e paisagem). No entanto, os dispositivos N95 e G705 possuem a função auto-rotação que permite que as aplicações sejam visualizadas nesses formatos.

No entanto, para o Teatro Móvel, os erros identificados no formato retrato foram críticos para o modo paisagem. Por exemplo, a aplicação apresentou um enorme espaço em branco na tela entre o nome “Descrição:” e o resumo da peça, no modo retrato foi possível visualizar uma parte do resumo, entretanto, no formato paisagem não foi possível visualizá-la sem o uso da barra de rolagem conforme pode ser observado na Figura 5.19.

Além disso, o usuário não identificava qual o nome da peça que ele lia o resumo. Por exemplo, a tela do resumo da peça não exibia o nome da peça que o usuário estava lendo, se ele clicasse no botão “voltar”, a aplicação voltava para a tela anterior que exibia uma lista com os nomes das peças em cartazes e, portanto, ele não poderia identificar o nome da peça que leu o resumo.

Requisito 12: resultado semelhante ao Requisito 12 da aplicação Shopping Móvel.

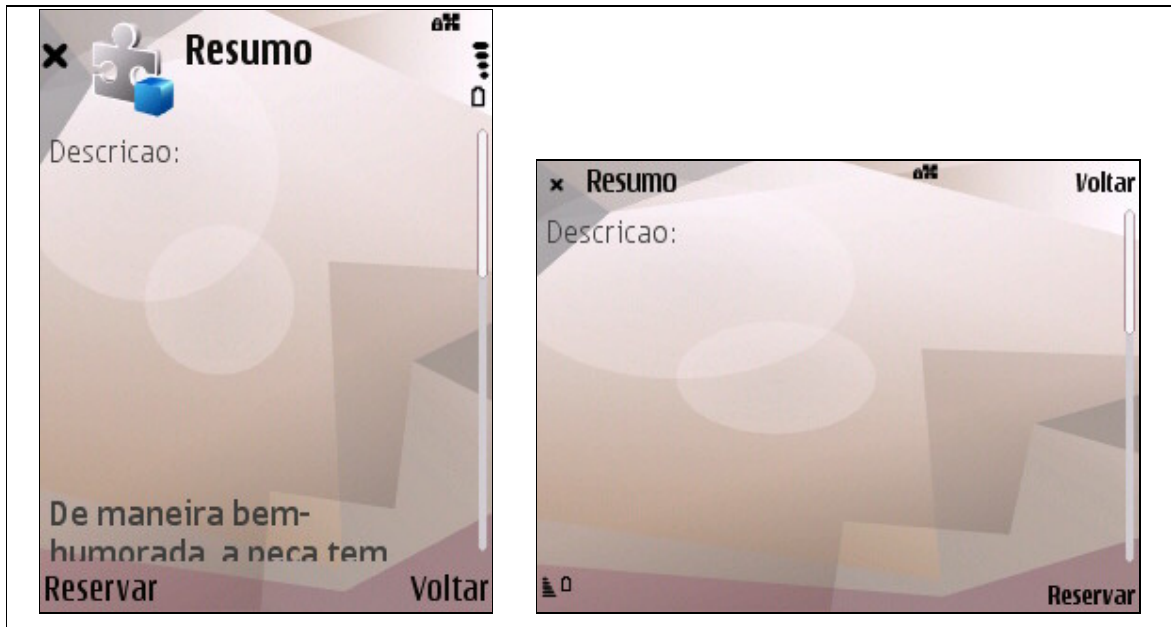


Figura 5.19– Resumo da peça visualizado respectivamente nos formatos retrato e paisagem

Requisito 13:

No WTK, o conteúdo da aplicação é visualizado sem cortes, sem o uso da barra de rolagem e com um bom contraste em relação ao plano de fundo a aplicação. Já nos demais emuladores, S60 e JP8, e nos seus dispositivos alvo, N95 e G705, a aplicação é visualizada com o uso da barra de rolagem e o plano de fundo é de acordo com o perfil escolhido no emulador e no dispositivo, dessa forma um bom contraste dependerá do perfil.

No S60 e no N95 que simularam comportamentos semelhantes, o texto da aplicação é exibido cortado e com três pontinhos (ver Figura 5.16) e um enorme espaço branco que dificulta a leitura (ver Figura 5.19), respectivamente dos nomes das peças e do seu resumo.

Já o JP8 e o G705 simularam comportamentos diferentes. Conforme ilustrado na Figura 5.20, o nome das peças com tamanho extenso é visualizado quebrado em várias linhas no JP8 e no G705, ele fica alternando na tela do dispositivo. Outro erro encontrado no JP8 e G705 é ilustrado na Figura 5.21, o rótulo para confirmar a reserva, “conf. reserva”, é exibido corretamente no N95, entretanto no JP8 aparece cortado e no G705 é mais crítico, sendo possível apenas visualizar “conf.”.

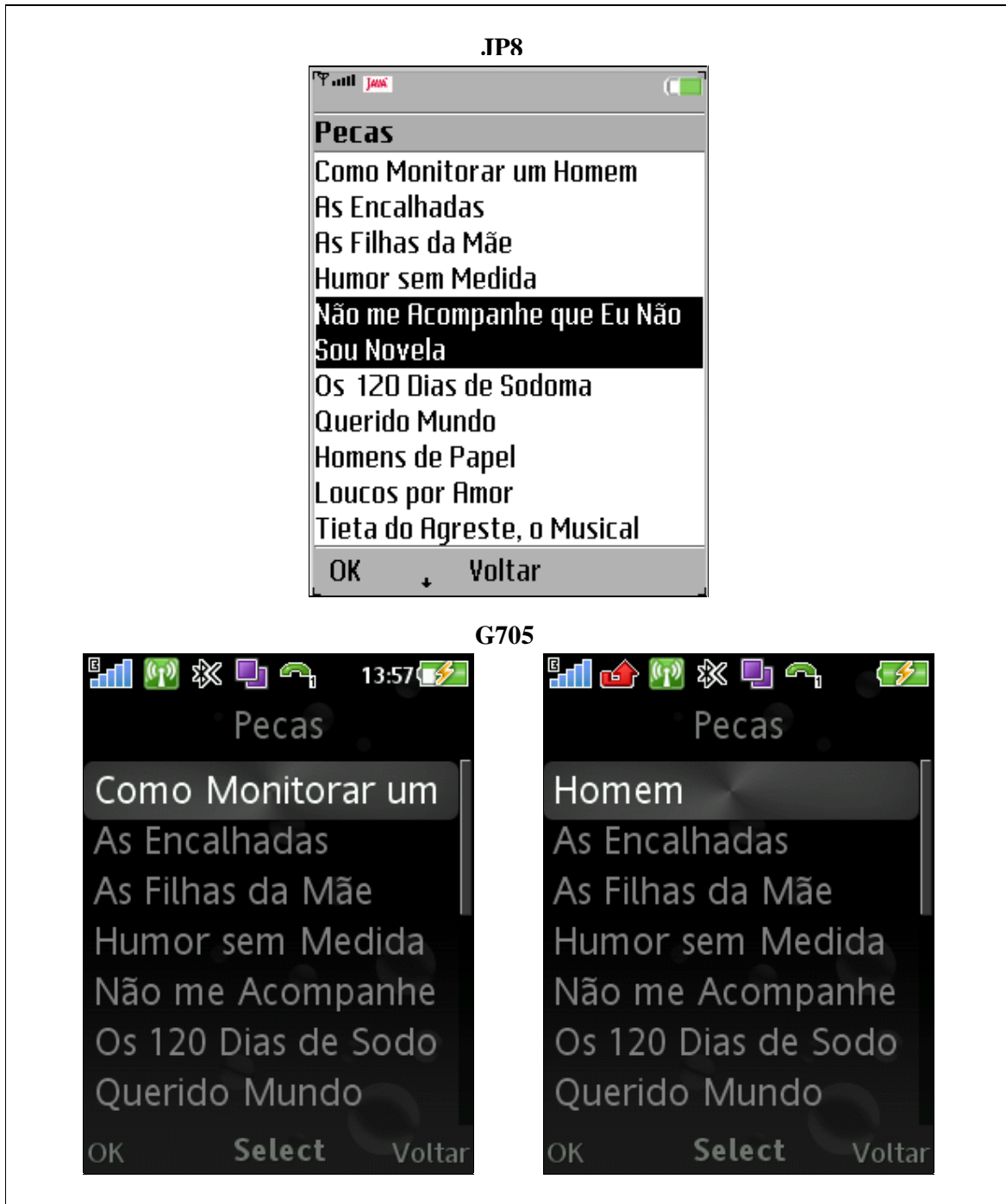


Figura 5.20– Visualização do texto no JP8 e no G705

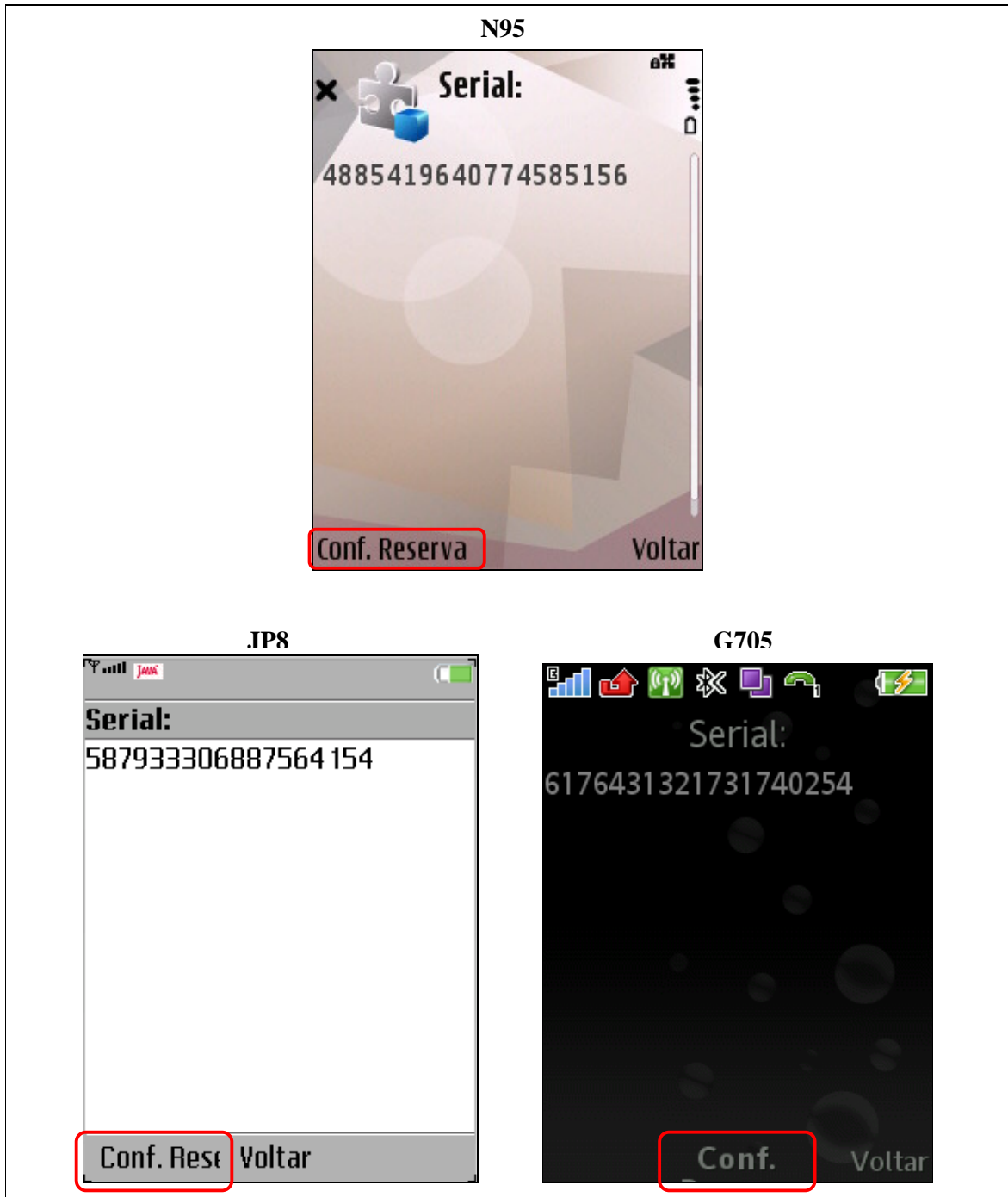


Figura 5.21 – Visualização do texto no JP8 e no G705

Requisito 14:

Quando o usuário vai reservar uma peça não era exibida na tela uma confirmação com o nome da peça e a quantidade de ingressos, apenas era exibido um número para o usuário confirmar a reserva. Além disso, se o usuário efetuou a reserva de uma peça e fizesse novamente a mesma, a aplicação não notificava o usuário que ele já possuía reserva para aquela peça.

Requisito 15: resultado semelhante ao Requisito 15 da aplicação Shopping Móvel.

Requisito 16:

Na estrutura do menu de acesso aos aplicativos do N95, as teclas *softkeys* da esquerda são usadas para ações positivas e as da direita para ações negativas. No Teatro Móvel as *softkeys* seguiram o padrão do N95 e, portanto, as *softkeys* da esquerda para as ações “Ok”, “reservar”, “cadastrar”, “conf. reserva” e “login” e as *softkeys* da direita para as ações “voltar” e “exit”.

O G705 além das teclas *softkeys* da esquerda para ações positivas e da direita para ações negativas também possui a tecla *softkey* do meio para ações positivas. O Teatro Móvel seguiu o padrão do dispositivo G705. Dessa forma, as teclas da esquerda para as ações “Ok”, “cadastrar” e “login” e as teclas da direita para ações “voltar” e “exit”. A tecla do meio também ficou com as ações positivas (e.g., “reservar”, “conf. reserva”, “Ok”, “edit” e “select”). Além disso, os rótulos “edit” e “select” na *softkeys* do meio que não são ações da aplicação, mas sim específicas do G705 conforme já discutido no Requisito 16 do Museu móvel.

5.3.4 Outras Aplicações

Requisito 1:

Conforme já mencionado neste capítulo, das dez equipes formadas, cinco conseguiram implementar a aplicação com sucesso. Dentre as quais, duas o modelo incremental, uma equipe o modelo de prototipação, uma o modelo espiral e uma equipe o XP com incremental. A Tabela 5.11 apresenta o modelo de processo utilizado pelas aplicações móveis e identifica se as equipes obtiveram sucesso na implementação.

O principal fator identificado para as equipes que não implementaram a aplicação foi o problema de comunicação entre o cliente (celular) e o servidor (*desktop* ou celular). Isso foi crítico para as equipes que utilizaram o modelo cascata e, por esse motivo, não conseguiram implementar a aplicação. Os testes, no modelo cascata, foram realizados depois da implementação e, portanto, erros só puderam ser identificados nas fases finais do ciclo.

Em virtude disso, acredita-se que requisitos importantes como a parte de comunicação da aplicação deveriam ter sido testados nas fases iniciais do ciclo de desenvolvimento e isso não foi possível com o modelo cascata. Já com os demais modelos, as equipes eram forçadas a entregar incrementos e protótipos que facilitou a identificação de erros mais cedo. Quanto ao XP, há indícios que pode ter sido adequado devido ao foco da metodologia ser na validação do projeto durante todo o processo de desenvolvimento. No entanto, a equipe utilizou o XP com o incremental, além de que era a única equipe que já possuía experiência no desenvolvimento de

aplicações móveis.

Tabela 5.9– Características do projeto de desenvolvimento das aplicações móveis

Aplicações Móveis	Modelo de Processo	Implementação
Sistema de Gerenciamento de Pedidos de Restaurante	Cascata	Não conseguiram implementar, apenas fizeram fluxo de telas no emulador
Controle de matrícula de estudantes universitários	Cascata	Implementaram, mas tiveram problemas de comunicação com o servidor
Sistemas de ingressos	Incremental	Implementaram
Controle Bancário	Incremental	Implementaram
Biólogo Móvel	Prototipação	Implementaram
Biblioteca Ambulante	Prototipação	Implementaram, mas tiveram problemas de comunicação de servidor
Sistema de Cadastro de Boletins de Ocorrência	Espiral com prototipação	Implementaram, mas tiveram problemas de comunicação de servidor
Mobile Hotel	Espiral	Implementaram, mas tiveram problemas de comunicação de servidor
IMBTalk - Instant Messenger Bluetooth	Espiral	Implementaram
Fotos Na Web	XP com Incremental	Implementaram

Com base nesses resultados, percebe-se que as equipes que utilizaram modelos de processo de desenvolvimento iterativo com foco na validação dos requisitos (i.e., prototipação, espiral e incremental) e, dessa forma, possibilitando os testes de serem realizados mais no início do desenvolvimento, obtiveram sucesso na implementação da aplicação. Das oito equipes que utilizam esses modelos, cinco implementaram a aplicação.

Requisito 2:

As equipes realizaram os testes nos seguintes ambientes: emulador genérico, emulador específico e dispositivo móvel.

Das dez equipes, cinco realizaram apenas testes no emulador tanto no genérico quanto no específico, quatro realizaram testes no emulador e no dispositivo móvel e uma equipe fez os testes apenas no dispositivo móvel.

Um erro constatado por seis equipes foi que as aplicações móveis funcionaram

corretamente no emulador, mas no dispositivo ocorreram problemas de comunicação com o servidor ao executá-las o que impossibilitou quatro equipes de realizarem os testes nos dispositivos móveis. O emulador utilizado por essas equipes durante os testes no ambiente *desktop* foi um emulador genérico em vez de um emulador específico para o dispositivo alvo.

Também foi observado pelas equipes erros de visualização da aplicação no dispositivo. A equipe Biólogo Móvel constatou uma diferença de exibição de menus ao testar aplicação no emulador específico. Por exemplo, o testador não efetuava o cadastro na aplicação devido a não conseguir visualizar na tela da aplicação a opção “concluir”. Esse fato ocorreu porque a equipe utilizou um emulador genérico para fazer os testes unitários da aplicação em vez de um emulador específico. Ao se testar a aplicação no emulador genérico não foram encontrados erros de visualização porque este possui uma tela maior do que o emulador específico e, portanto, apresenta uma maior quantidade de informação na tela.

Dessa forma, o uso de emuladores genéricos durante o desenvolvimento pode não identificar erros que ocorrem nos emuladores específicos. Além disso, erros de comunicação podem não ser identificados nos emuladores específicos e, portanto, os testes devem ser realizados também nos dispositivos móveis.

Requisitos 3:

Das dez equipes de Engenharia de Software foram gerados treze Relatórios de Testes incluindo os testes feitos no emulador e no dispositivo alvo. Os relatórios possuíam campos conforme as informações contidas neste requisito (e.g., versão do emulador e/ou dispositivo e ambiente de teste) e facilitou a identificação de erros. No entanto, quatro equipes não preencheram o campo “versão do emulador e/ou dispositivo” do Relatório de Testes o que impossibilitou uma análise dos erros encontrados referente ao local onde os testes foram realizados.

Requisito 4:

As informações para reportar os erros (e.g., descrição do erro, passo a passo e frequência) foram em conformidade com o Requisito 4. No entanto, as equipes não preencheram todas as informações e três não documentaram nenhum erro. Apenas duas equipes colocaram todas as informações solicitadas sobre o erro reportado e, com isso, facilitou bastante a análise do erro.

Além disso, uma equipe acrescentou o campo “*status*” ao erro reportado. Isso aconteceu porque os alunos receberam o modelo de relatório no início do desenvolvimento, e conforme encontravam os erros, reportavam no relatório também com os erros identificados e corrigidos.

Requisito 5:

Das dez equipes, sete equipes não testaram este requisito. Acredita-se que devido à falta de experiência das equipes, as mesmas se concentraram em implementar os requisitos funcionais da aplicação e, portanto, não consideraram as limitações dos dispositivos móveis.

Além disso, as três equipes restantes identificaram erros de visualização da aplicação na tela do dispositivo que é menor que o emulador. Entretanto, essas equipes não reportaram dados suficientes para uma análise das demais limitações.

Requisito 6:

Somente as cinco equipes que conseguiram implementar testaram as aplicações de acordo com este requisito e os resultados foram satisfatórios. Os aplicativos do dispositivo móvel continuaram funcionando corretamente após a instalação da aplicação móvel.

Requisito 7:

As equipes somente realizaram o teste em laboratório utilizando o emulador e/ou dispositivo. No emulador foram identificados erros de funcionalidade e de usabilidade. Por exemplo, a equipe que desenvolveu a aplicação “Sistema de Gerenciamento de Pedidos de Restaurante” testou o fluxo das telas no emulador. Já no dispositivo foram encontrados erros de comunicação entre o cliente e o servidor, além de erros de visualização da aplicação na tela do dispositivo.

A aplicação “Fotos na web” foi testada somente no dispositivo móvel e em laboratório, pois segundo a equipe, a aplicação precisava das funcionalidades câmera e GPS que não eram possíveis simular no emulador. No entanto, acredita-se que questões de usabilidade, tais como a estrutura do menu e a usabilidade das telas da aplicação poderiam ter sido testadas no emulador em laboratório. No entanto, percebe-se que a natureza dos requisitos funcionais da aplicação “Fotos na web” (e.g., GPS) pode ser afetada por questões de mobilidade e, portanto, eles deveriam ter sido testados em campo.

Requisito 8:

As equipes não realizaram teste de usabilidade. No entanto, os requisitos propostos que tratam de usabilidade das aplicações móveis (requisitos de nove a dezesseis apresentados no Capítulo 4) foram testados desde o início do desenvolvimento pelas equipes. Além disso, através do rascunho de telas (do inglês *screen sketch*⁵), a equipe que desenvolveu o Sistema de

⁵ *Screen Sketch*: desenho simplificado da aparência do sistema. Utilizados pelos especialistas de IHC (Interface Homem-Máquina)

Gerenciamento de Pedidos de Restaurante testou o fluxo de telas da aplicação.

Requisito 9:

As equipes relataram que foram encontrados erros que prejudicaram a interação do usuário com a aplicação, os quais foram:

- A ausência do rótulo “voltar” nas telas;
- Muitas informações na tela dificultaram a leitura e compreensão dos dados.;
- Não visualização de rótulos dos botões (e.g., concluir) da aplicação na tela do dispositivo.

Requisito 10:

Este requisito foi testado por seis equipes e todas relataram que a aplicação manteve seu estado após as interrupções.

Embora os emuladores utilizados (e.g, S60 e JP8) por três equipes permitissem testar algumas interrupções (e.g., pausar a aplicação, estado da bateria e notificação de alarme), as equipes relataram que testar este requisito no emulador era inviável.

Requisito 11:

As aplicações móveis desenvolvidas pelas equipes não deram suporte aos dois formatos de tela. No entanto, os dispositivos móveis que davam o suporte para os dois formatos (e.g., N95, Xperia e C905) apresentaram resultados divergentes. Por exemplo, a aplicação Sistema de ingressos foi visualizada corretamente no modo retrato e paisagem no seu dispositivo alvo, Xperia. Já a aplicação “IMBTalk” só foi visualizada no formato retrato no C905, embora este dispositivo também fornecesse o suporte para o modo paisagem.

Além disso, um fator identificado foi que a equipe que desenvolveu a aplicação “Fotos na Web” não testou se a aplicação se adaptava ao modo retrato e paisagem. Um dos requisitos funcionais da aplicação é “tirar” fotos através da câmera do dispositivo, salvá-las e enviá-las para um *site* que armazena fotos (e.g., Flickr). As fotos tiradas pela câmera do dispositivo através da aplicação podem ser enviadas no formato que o usuário tirou (e.g., retrato e paisagem) no dispositivo alvo da aplicação, N95. No entanto, não foi testado pela equipe se as fotos tiradas nos formato desejado eram mantidas pela aplicação ao ser enviada para armazenamento.

Requisito 12:

As aplicações móveis desenvolvidas pelas equipes são cliente e servidor e, portanto,

precisam se comunicar com a rede, por exemplo, através do Wi-Fi, GPRS ou 3G. No entanto, segundo as equipes, a permissão para estabelecer conexões com a rede foi pedida pelo próprio dispositivo móvel. Esse resultado é semelhante ao Requisito 12 da aplicação Shopping Móvel, Museu Móvel e Teatro Móvel.

Requisito 13:

Não foram reportados erros referentes a este requisito. No entanto, as equipes relataram o uso de texto justificado e em caixa alta. As aplicações que utilizaram imagens apareceram cortadas ou distorcidas no dispositivo alvo.

As aplicações móveis só apresentaram barra de rolagem quando necessário, com exceção da aplicação “IMBTalk” que é um bate-papo via Bluetooth. Nessa aplicação, o uso da barra depende do tamanho da mensagem enviada pelo usuário e esta é limitada pelo próprio dispositivo móvel. Por esse motivo, segundo o relato da equipe, as mensagens muito extensas eram difíceis de ler devido ao uso da barra de rolagem.

Outro fator observado é que a Biblioteca Ambulante foi a única aplicação que apresentou um plano de fundo.

Acredita-se que utilizar este requisito nas fases iniciais do ciclo ajudou os desenvolvedores a melhorar a usabilidade das aplicações móveis e, portanto, percebe-se uma melhoria na apresentação das telas das aplicações móveis.

Requisito 14:

As equipes não testaram este requisito. Além disso, a equipe que desenvolveu a aplicação “IBMTalk” relatou que este requisito não se aplicava ao bate-papo. No entanto, a ação “sair do bate-papo” é uma ação não reversível e a equipe não tratou essa questão. Dessa forma, acredita-se que erros de usabilidade, como por exemplo, ausência de uma mensagem de confirmação para uma ação não reversível, poderiam ter sido identificados em virtude deste requisito.

Requisito 15:

Nenhuma das dez aplicações desenvolvidas pelos equipes de Engenharia de Software utilizaram som.

Requisito 16:

Em geral, as teclas *softkeys* utilizadas para acessar as funções das aplicações móveis se adaptaram ao dispositivo alvo. No entanto, as equipes Mobile Hotel e Sistema de Cadastro de Boletim de Ocorrência relataram que os rótulos das teclas *softkeys* da esquerda das suas aplicações

apresentaram tanto ações positivas quanto negativas.

A equipe que desenvolveu a aplicação Sistema de Ingressos para o dispositivo alvo Xperia observou que o acesso aos rótulos das teclas *softkeys* da esquerda e direita também poderiam ser acessadas através da função “*touch screen*” do dispositivo.

Além disso, um erro reportado pela equipe que desenvolveu “Fotos da web” foi que o botão da câmera utilizado para tirar as fotos da aplicação não era o mesmo botão utilizado pela câmera do dispositivo e isso, muitas vezes, atrapalhava o usuário no uso da aplicação e a mesma não tratou essa questão.

5.4 Discussão dos Resultados Alcançados

Em primeiro lugar, na análise dos resultados obtidos com a utilização dos requisitos propostos nos testes das três aplicações móveis (i.e., Shopping Móvel, Museu Móvel e Teatro Móvel) foi que o testador experiente no conhecimento dos requisitos propostos identificou mais erros que o testador inexperiente. Mesmo assim, o testador inexperiente utilizando os requisitos elicitados encontrou erros que não foram identificados pelo testador experiente.

Já na disciplina de Engenharia de Software em 2009, foi constatado que equipes se concentraram mais em testar os requisitos funcionais em vez das questões específicas do ambiente móvel. Esse fato ocorreu devido à falta de experiência das equipes em desenvolver aplicações móveis. Apenas uma equipe possuía experiência nesse tipo de desenvolvimento. Em virtude disso, as informações contidas nos relatórios de teste não foram completas e isso dificultou a análise dos resultados. Os resultados apresentados na subseção 5.3.4 foram baseados nos relatórios de testes gerados pelas equipes e, portanto, não foram verificados, mediante execução das aplicações, os erros reportados

Um fator observado, em relação à disciplina, foi que as equipes que não obtiveram sucesso, em executar a aplicação no dispositivo móvel devido à problemas de comunicação com o servidor, realizaram os testes e identificaram erros nos emuladores com uso dos requisitos propostos. No entanto, determinados requisitos que podiam ter sido testados no emulador, as equipes não consideraram viável testar nesse ambiente, como, por exemplo, o Requisito 10.

Outro fator identificado é que as equipes não relacionaram os erros reportados aos requisitos. Por exemplo, ao testar a aplicação de acordo com o requisito proposto, a equipe colocava como resultado obtido “satisfatório”, entretanto, no mesmo relatório era reportado um erro identificado através desse requisito. O motivo para tal situação é a falta de experiência dos alunos na documentação de artefatos, uma vez que o primeiro contato em elaborar

documentação, é feito na disciplina de Engenharia de Software.

Em relação ao modelo de processo de desenvolvido abordado no Requisito 1, a sua utilização só foi analisada nas dez aplicações desenvolvidas na disciplina de ES em 2009. Embora as três aplicações móveis desenvolvidas pelos alunos em 2006 tenham utilizado o modelo cascata, não foi feita uma análise devido a não utilização dos requisitos propostos durante o desenvolvimento das mesmas. Já na disciplina de ES em 2009, os alunos que utilizaram uma abordagem (e.g., incremental, espiral e prototipação) com foco no processo de teste obtiveram sucesso no desenvolvimento das suas aplicações. Além disso, todas as equipes que utilizaram o modelo incremental conseguiram implementar a aplicação.

Quanto ao Requisito 2, pode-se afirmar que nos três ambientes (i.e., emulador genérico, emulador específico e dispositivo móvel) puderam ser identificados erros de funcionalidade e de usabilidade não relacionados com a visualização da aplicação na tela. Já os erros de usabilidade, em relação à visualização da aplicação na tela, se comportam de maneiras diferentes e, portanto, depende do ambiente testado. No emulador específico foram encontrados mais erros do que no emulador genérico e determinados erros só foram identificados no dispositivo móvel. Além disso, a mesma aplicação se comportou de maneira diferente em cada modelo de dispositivo móvel utilizado, e erros críticos só ocorreram em um determinado modelo de dispositivo. Por exemplo, as imagens da aplicação Museu Móvel que foram visualizadas distorcidas e cortadas no N95, entretanto, foram exibidas sem cortes no G705 (ilustrado na Figura 5.9).

A Tabela 5.9 resume as principais características testadas nos testes feitos no emulador genérico (WTK), no emulador específico (S60 e JP8) e no dispositivo móvel (N95 e G705).

Os requisitos 3 e 4 ajudaram a identificar os erros e em qual ambiente ele ocorria. Os modelos de relatório de teste utilizados pelas equipes foram semelhantes ao modelo utilizado pelos testadores nos testes das três aplicações móveis. No entanto, houve apenas uma modificação com o intuito de identificar em que fase do ciclo de desenvolvimento os requisitos propostos poderiam ser utilizados. Dos treze relatórios entregues, apenas duas equipes preencheram o campo solicitado e, portanto, os dados não foram suficientes para uma análise.

Tabela 5.10– Características testadas nos Emuladores Genéricos, Específicos e nos Celulares

Características testadas	Emulador Genérico WTK	Emuladores Específicos S60 e JP8	Celulares N95 e G705
Fonte do texto	Maior	Menor que no Genérico	Menor que no específico
Processamento	Rápido	Lento	Mais Lento
Tela	Maior	Menor que o Genérico	Menor que o específico
Memória	Monitor de Memória com suporte a limitação.	Monitor de Memória com suporte a limitação.	Necessário suporte da aplicação ou ferramenta que faça medição no dispositivo alvo.
Capacidade de Armazenamento	Não se aplica	Não se aplica	Se aplica
Bateria	Não se aplica	Não se aplica	Medição do consumo real
Largura de Banda	Não se aplica	Não se aplica	Wi-Fi , GPRS ou 3G
Mobilidade	Pausar a aplicação	No S60 é possível simular: eventos básicos (status da bateria), notificações (alarme), mensagens (SMS, MMS), Bluetooth, IrDA, dentre outros. JP8 simula apenas um para a execução da aplicação através da função <i>pause</i> .	As interrupções em laboratório.
Ergonomia	<i>Design</i> do teclado parecido com de um celular qualquer, contudo não simula o comportamento real do teclado de um celular real. Não suporta características de selecionar, copiar/recortar e colar. Não utiliza barra de rolagem.	<i>Design</i> do teclado muito semelhante ao dispositivo alvo e simula o comportamento bem próximo ao celular alvo. Suporta características de selecionar, copiar/recortar/colar embora nem sempre o dispositivo alvo as suporte. S60 utiliza barra de rolagem, mas o JP8 não utiliza barra de rolagem.	S60 suporta as características de selecionar, copiar/recortar/colar e o N95 não suporta. Já o JP8 não suporta, mas o G705 suporta. Sempre utiliza barra de rolagem.

Em relação às limitações dos dispositivos móveis abordadas nos requisito 5, percebe-se que a natureza da aplicação também influencia no seu comportamento. Por exemplo, o processamento do Teatro Móvel, que possui basicamente informações textuais, em relação ao tempo de resposta para executar uma tarefa, foi mais rápido do que o do Shopping Móvel e o do Museu móvel, que possuem imagens. Em virtude disso, a aplicação consumia pouca memória RAM e, dessa forma, não foram identificados erros relacionados ao consumo máximo de memória do dispositivo como foi observado no Museu Móvel. Além disso, a capacidade de

armazenamento é mais crítica quando a aplicação precisa salvar dados nas memórias internas dos dispositivos móveis, entretanto, nenhuma das três aplicações testadas possui tal requisito funcional.

Todas as aplicações que foram testadas com o uso do Requisito 6, não interferiram no comportamento das aplicações já instaladas nos dispositivos móveis.

Em relação ao Requisito 7, que trata das características testadas em laboratório e em campo. A Tabela 5.9 apresenta um resumo das funcionalidades testadas em laboratório em cada ambiente de teste (i.e., emulador e dispositivo).

Tabela 5.11– Características testadas em laboratório

Ambiente de Teste	Modo de teste	Funcionalidades Testadas	Limitações Testadas
Laboratório	Emulador	Requisitos Funcionais que não dependem do meio externo. Questões de usabilidade, tais como, estrutura de menu, ausência de mensagens, imagens cortadas.	Memória, Processamento, Tela e Interrupção.
	Celular	Requisitos Funcionais que não dependem do meio externo. Todas as questões de usabilidade que não envolvem o meio externo.	Memória, tela, bateria, processamento, capacidade de armazenamento, largura de banda e interrupções.

Outro fator observado é que o teste em laboratório quando feito no dispositivo móvel identifica mais erros do que feito no emulador. Dessa forma, como não foram feitos testes em campo com o dispositivo, acredita-se que muitos aspectos de usabilidade (e.g., iluminação, barulho e movimento físico) do mundo real deixaram de ser verificados durante os testes. Além disso, questões de desempenho da comunicação com a rede no mundo real (e.g., GPRS e 3G) que poderiam falhar se testados em campo, não foram tratados e, portanto, erros podem não ter sido identificados.

Em relação ao Requisito 8, que inclui o teste de usabilidade no ciclo de desenvolvimento, erros de usabilidade foram identificados e apresentados nos requisitos de 9 a 16 nas subseções 5.3.1, 5.3.2, 5.3.3 e 5.3.4. Além disso, as equipes com experiência otimizaram o desenvolvimento das aplicações móveis para evitar que o consumo de recursos dos dispositivos afetasse a usabilidade e até mesmo o funcionamento da aplicação, como foi o caso do Shopping Móvel.

Um fator observado foi que as equipes de Engenharia de Software em 2009 que utilizaram os requisitos propostos para o teste de usabilidade identificaram menos erros de

usabilidade que poderiam ser identificados através do uso dos Requisitos 9 e 13.

Em relação ao Requisito 9, que trata das ações que podem ser tomadas para o usuário se localizar na aplicação, os principais erros constatados foram a dificuldade de visualização das mensagens na aplicação e o não uso do rótulo “voltar”. Acredita-se que esses erros sejam críticos para que o usuário saiba que ação tomar na aplicação.

Quanto ao Requisito 10, todas as aplicações que foram testadas de acordo com este requisito mantiveram o seu estado após as interrupções simuladas.

Nenhuma das aplicações testadas forneceu suporte para os dois formatos retrato e paisagem conforme o Requisito 11. No entanto, foi observado que determinados modelos de dispositivos (e.g., N95, G705, XPeria e C905) já permitem visualizar as aplicações nos dois formatos. No entanto, mesmo que o dispositivo móvel forneça o suporte, isso não garante que a aplicação seja visualizada corretamente nos dois formatos e, portanto, quando o suporte é dado pelo dispositivo alvo, a aplicação deve ser testada no modo retrato e paisagem.

Em relação ao Requisito 12, todas as aplicações testadas pediram inicialmente para estabelecer comunicação com a rede. Porém, como as solicitações de comunicação dependem de como as políticas de segurança do ambiente (e.g., toolkit ou KVM – Kilo Virtual Machine) exigidas ou não pela aplicação foram configuradas pelo desenvolvedor ou pelo fabricante do dispositivo móvel. Acredita-se que esse fato ocorreu devido às aplicações serem desenvolvidas em JME e, dessa forma, as solicitações de segurança são configuradas, a princípio, para exigirem permissão para a solicitação de comunicação para as aplicações não certificadas.

O Requisito 13 ajudou as equipes de ES de 2009 a identificarem erros logo no início do desenvolvimento e, dessa forma, erros não foram reportados. Além disso, foi observado uma melhoria no desenvolvimento das interfaces das aplicações em relação à usabilidade. Acredita-se que a utilização dos requisitos propostos ajudou a identificar erros de usabilidade mais cedo diferente do que aconteceu nas três aplicações desenvolvidas em 2006 que não utilizaram os requisitos elicitados nesta dissertação e, portanto, foram identificados mais erros através do Requisito 13.

O Requisito 14 não foi utilizado por nenhuma das equipes de ES em 2009 apesar de aplicações como o “IMBTalk” conter ações não reversíveis e que são críticas para a aplicação. Além disso, esse resultado foi semelhante ao Shopping Móvel e ao Teatro Móvel. Acredita-se que isso ocorreu devido às equipes não saberem diferenciar o que seriam as ações não reversíveis das suas aplicações.

Quanto ao Requisito 15, nenhuma das treze aplicações testadas possuíam som. Acredita-se que implementar o som em uma aplicação móvel não seja uma tarefa tão trivial para que equipes sem experiência no desenvolvimento de aplicações móveis possam se preocupar a princípio. O que foi observado é que as equipes se concentram em desenvolver as funcionalidades básicas da aplicação. Dessa forma, o não uso do som prejudicou a usabilidade da aplicação principalmente nas mensagens de erro.

Em relação às teclas *softkeys*, foi observado que, embora elas sejam o principal acesso do usuário para interagir com a aplicação, os erros decorrentes das *softkeys* não seguirem o padrão do dispositivo não foram corrigidos pelas equipes de ES em 2009 durante o desenvolvimento das aplicações.

5.5 Conclusão

Este capítulo apresentou os resultados obtidos através da aplicação dos requisitos no teste de três aplicações móveis desenvolvidas na disciplina de ES em 2006 e durante o desenvolvimento de aplicações móveis na disciplina de ES em 2009. Para isso, foi elaborado um modelo de relatório de teste para ser utilizado pelos testadores nos testes das três aplicações e pelos alunos durante os testes das aplicações móveis. O relatório continha os requisitos propostos apresentados no Capítulo 4 e também informações adicionais com o intuito de auxiliar na sua análise.

Com o objetivo de tornar o ambiente de teste mais completo foi utilizado diferentes tipos de dispositivos móveis, emuladores e testadores (e.g., experiente e inexperiente). Além disso, os requisitos foram utilizados em aplicações móveis que já haviam sido desenvolvidas anteriormente e também durante o desenvolvimento de novas aplicações móveis.

Com base na análise apresentada neste capítulo pode-se afirmar que com o uso dos requisitos propostos foi possível identificar tanto problemas intrínsecos ao contexto móvel quanto problemas que não são característicos deste ambiente, mas que nele se tornam mais complexos e desafiadores para o processo de teste e usabilidade das aplicações móveis. Dessa forma, o uso dos requisitos propostos nos testes de aplicações móveis se mostrou eficaz na identificação de erros e, portanto, propicia uma melhoria na qualidade das aplicações móveis desenvolvidas.

No próximo capítulo são apresentadas as conclusões e as principais contribuições desta dissertação, além dos possíveis trabalhos futuros.

Capítulo 6

Conclusão e Trabalhos Futuros

Esta dissertação propôs um conjunto de requisitos para o processo de teste e usabilidade de aplicações móveis. O objetivo principal dos requisitos propostos é auxiliar a equipe de testes de aplicações móveis a identificar erros inerentes ao ambiente móvel e, dessa forma, suprir as deficiências do processo de teste tradicional. Para a validação dos requisitos propostos, aplicações móveis foram testadas e foi uma análise detalhada dos erros encontrados.

O restante deste capítulo apresenta as principais contribuições na Seção 6.1 e os trabalhos futuros que podem ser derivados desta dissertação na Seção 6.2.

6.1 Contribuições

A mobilidade juntamente com as limitações dos próprios dispositivos impactam no processo de desenvolvimento das aplicações móveis. Por exemplo, o processo de teste tradicional falha nessas questões intrínsecas ao ambiente móvel, surgindo daí a necessidade de adaptá-lo para tratar dessas questões específicas para as aplicações móveis.

Nesse contexto, a principal contribuição deste trabalho é um conjunto de requisitos elicitados para o teste de aplicações móveis, o qual foca principalmente nas limitações específicas do ambiente móvel e atua como um guia para a equipe de testes dessas aplicações.

Os requisitos propostos nesta dissertação resultam de um levantamento bibliográfico aliado a experiências práticas de desenvolvedores de aplicações móveis na etapa de testes. Esses requisitos são em parte genéricos o suficiente para serem utilizados em aplicações *desktops* e em parte específicos para aplicações executando em dispositivos móveis. Sendo assim, nesse conjunto são tratadas além das limitações do ambiente móvel, as limitações do ambiente *desktop* e as do próprio desenvolvedor, que não possui muita experiência em desenvolver aplicações móveis.

Para a elicitação dos requisitos propostos foi utilizada uma metodologia que também é considerada uma contribuição deste trabalho. Em resumo, ela é composta das seguintes fases:

- O *levantamento teórico-prático* consistiu de uma extensa revisão bibliográfica na literatura.
- A *coleta de dados* foi feita através de entrevistas pessoais e aplicação de questionários com profissionais que possuem experiência no desenvolvimento de aplicações móveis.
- A *análise dos dados* consistiu no levantamento da porcentagem de uso dos requisitos propostos pelos profissionais, além de investigar questões relacionadas com o processo de teste das aplicações móveis.
- A *validação* foi realizada em duas etapas. Na primeira etapa, os requisitos propostos foram utilizados por dois testadores, um experiente e outro inexperiente, no teste de três aplicações móveis desenvolvidas e já testadas por alunos da disciplina de ES no semestre de 2006.1. A segunda etapa consistiu na utilização dos requisitos propostos durante todo o ciclo de desenvolvimento por alunos de graduação e pós-graduação da disciplina de Engenharia de Software no semestre de 2009.1.

No levantamento bibliográfico observou-se uma preocupação dos pesquisadores em propor mudanças no processo de desenvolvimento para que o mesmo acomodasse as questões do ambiente móvel. No entanto, durante as entrevistas realizadas na fase de coleta de dados foi observado que o desenvolvimento das aplicações móveis é muitas vezes tratado como desenvolvimento de aplicações *desktop* e as empresas que atuam na área não utilizam um processo de teste específico que trate as questões inerentes do ambiente móvel.

Já a fase de análise dos dados ajudou a refinar os requisitos já elicitados na literatura, além de obter um consenso de quais requisitos são fundamentais para o processo de teste e usabilidade das aplicações móveis, de acordo com os resultados das entrevistas realizadas na fase anterior.

Durante a validação, com a utilização dos requisitos propostos para o teste de três aplicações móveis, Shopping Móvel, Museu Móvel e Teatro Móvel, conforme detalhado no Capítulo 5, foi possível identificar novos erros nessas aplicações, as quais já tinham sido testadas pelos alunos durante a disciplina de ES no semestre 2006.1. Além disso, quando esses requisitos foram utilizados no desenvolvimento de aplicações móveis na disciplina de ES do semestre 2009.1, eles ajudaram a guiar os desenvolvedores na descoberta de erros nas etapas iniciais do processo de desenvolvimento.

É importante ressaltar ainda que os requisitos propostos para o teste de usabilidade incentivou os alunos a tratarem questões de usabilidade do ambiente móvel e a utilizar artifícios

de IHC (Interface Homem Máquina), como rascunho de telas que ajuda a identificar o fluxo de telas da aplicação.

Com base nos resultados da fase de validação da metodologia utilizada nesta dissertação, foi constatado que os requisitos elicitados ajudaram a encontrar erros nas aplicações móveis mesmo quando utilizados por testadores inexperientes. Além disso, observou-se que o uso dos requisitos propostos, ao identificar mais erros nas aplicações móveis testadas, propicia uma melhoria do processo de teste e, portanto, na qualidade final das aplicações móveis geradas.

Os requisitos propostos, portanto, são adequados para o processo de teste de aplicações móveis, pois adicionam características que tratam mobilidade, limitações e fatores de usabilidade, os quais são intrínsecos ao ambiente móvel e com isso ajudam a suprir deficiências dos processos de teste tradicionais.

6.2 Trabalhos Futuros

As possíveis linhas de pesquisa para a continuação e aprimoramento deste trabalho são:

- Comparar resultados de testes com a utilização dos requisitos propostos

O próximo passo para este trabalho é comparar os resultados obtidos nos testes de aplicações móveis sem a utilização dos requisitos elicitados e os resultados obtidos com a utilização dos requisitos propostos.

- Elicitar requisitos específicos para o teste de determinados tipos de aplicações móveis

Os requisitos elicitados nesta dissertação não dependem do tipo de aplicação móvel. No entanto, durante o desenvolvimento desta dissertação foram identificados requisitos específicos para determinados tipos de aplicações móveis, tais como jogos e *web*, as quais possuem características bem particulares que devem ser abordadas no processo de teste. Dessa forma, um possível trabalho futuro desta dissertação é elicitar requisitos específicos para determinado tipo de aplicação móvel (e.g., jogos, *web*, corporativas, *e-commerce*).

- Sistematização dos requisitos propostos

Os requisitos elicitados foram apresentados independentes de um processo de desenvolvimento e/ou de teste específico. No entanto, para ilustrar como os requisitos podem ser aplicados em um processo, eles podem ser incorporados a uma disciplina de teste de um processo bem estabelecido (e.g., RUP) se adaptando a artefatos de testes existentes ou mesmo propor novos artefatos, além de uma identificação de responsáveis e de possíveis sequências para as realizações das atividades.

- Propor um modelo de processo de teste específico para aplicações móveis

A sistematização dos requisitos propostos em um modelo de processo pode gerar novos artefatos que poderão ser insumos para a proposta de um modelo de processo de teste específico para aplicações móveis tratando todas as questões inerentes do ambiente móvel.

Referências Bibliográficas

- Almeida, E. S.; Alvaro, A.; Garcia, V. C.; Mascena, J. C. C. P.; Burégio, V. A. A.; Nascimento, L. M.; Lucrédio, D e Meira, S. R. L. (2007). C.R.U.I.S.E: Component Reuse in Software Engineering, C.E.S.A.R e-book, Brazil, 2007. Disponível em: <<http://www.rise.com.br/research/publications.htm>>. Acesso em: 11 mar. 2008.
- Andrade, R. M. C. and Logrippo, L. (2006). Morar: A pattern language for mobility and radio resource management. In Dragos Manusecu, Markus Völter, J. N., editor, Pattern Language of Program Design 5, chapter 10, pages 213_256. Addison-Wesley.
- Andrade, R. M. C.; Marinho, F. G.; Leitão, V. L. e Rocha, Lincoln S. (2008). Uma proposta de Metodologia para o Ensino de Engenharia de Software. In: Fórum de Educação em Engenharia de Software - FEES 2008, 2008, Campinas. Monografia em Ciência da Computação nº 43/08 - Fórum de Educação em Engenharia de Software. Rio de Janeiro: PUC - Pontifícia Universidade Católica do Rio de Janeiro, 2008. v. 43. p. 107-115.
- Ballard, B. (2007). Designing the Mobile User Experience. Little Springs Design, Inc., USA: Wiley, 2007.
- Barreto, J. (2009). Celular LG se recusa a discar 911. Abril: Info Plantão, 03 de mar. 2009. Disponível em: <<http://info.abril.uol.com.br/aberto/infonews/032009/04032009-10.shl>>. Acesso em: 17 ago. 2009.
- Barnes, S. J. (2003). The wireless application protocol: strategic implications for wireless internet services. In Mobile Commerce: Technology, theory, and Applications, B. E. Mennecke and T. J. Strader, Eds. IGI Publishing, Hershey, PA, 145-161.
- Bastos, A.; Rios, E. Cristalli, R. e Moreira, T. (2007). Base de Conhecimento em Teste de Software. São Paulo: Martins, 2 ed. rev. 2007.
- Betioli, A. H. (2004). Avaliação de usabilidade para os computadores de mão: Um estudo comparativo entre três abordagens para ensaios de interação. Florianópolis: UFSC, 2004. 210 p. Tese de Doutorado, Doutorado em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2004.

- BugHuntress QA Lab (2007). Mobile Usability Testing Problem and Solutions. In: Proceedings of the Conference “Quality Assurance: Management & Technologies” (QAMT Ukraine’07), 2007.
- BugHuntress (2009). Bug Huntress - each found bug is a step to success of your software in the market. Disponível em: <<http://www.bughuntress.com/>>. Acesso em: 24 ago. 2009.
- Cai, J. e Goodman, D. J. (1997) “General Packet Radio Service in GSM,” IEEE Comm. Magazine, pp. 122-131, Oct. 1997.
- Collins, D. e Smith, C. (2002). 3G Wireless Networks. USA: McGraw - Hill, 2002.
- Craig, R. D. e Jaskiel S. P. (2002). Systematic Software Testing. Boston: Artech House, 2002.
- Delamaro, M. E., Maldonado, J. C. e Jino, M. (2007). “Introdução ao teste de software”. Rio de Janeiro: Elseiver, 2007. 384p.
- Dijkstra, E. W., Dahl, O.J. e Hoare, C.A. (1972). Structured programming. Londres: Academic Press.
- Fernandes, T. e Dias, A. (2006). Processo de Testes para Desenvolvimento de Jogos Celulares. In: Proceedings of the V Brazilian Symposium on Computer Games and Digital Entertainment (SBGames Computing). Short Paper. 2006.
- Forum.Nokia.com (2009). S60 Platform SDK 3rd Edition, FP 1. Disponível em: <http://www.forum.nokia.com/Tools_Docs_and_Code/Tools/Platforms/>. Acesso em: 20 ago. 2009.
- Graham, D. (2002) Requirements and Testing: Seven Missing-Link Myths. IEEE software, vol. 19, no. 5, 2002, p. 15-17.
- McNamara, N. e Kirakowski, J. (2006). Functionality, usability, and user experience: three areas of concern. Journal Interactions, vol. 13, n.6, 2006, pp 26-28.
- IEEE (2004). Guide to the Software Engineering Body of Knowledge. IEEE Computer Society, 2004.
- IEEE Computer Society (2007). “Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.” IEEE Std. 802.11-2007

- Inthurn, C. (2001). *Qualidade & Teste de Software*. Florianópolis: Visual Books, 2001.
- Kaikkonen, A; Kekäläinen, A Cankar, M; Kallio, T. e Kankainen, A. (2005). Usability testing of mobile applications: A comparison between laboratory and field testing. *Journal of Usability Studies*, vol.1, 2005, p.4-17.
- Kaikkonen, A. e Roto, V. (2003) Navigating in a Mobile XHTML Application. In: *Proceedings of the Conference on Human Factors in Computing Systems (SIGCHI 2003)*, Florida, USA, 2003, pp. 329-336.
- Kjeldskov, J. e Graham, C. (2003). A Review of Mobile HCI Research methods, In: *Proceedings of the 5th International Mobile Conference (HCI 2003)*, Udine, Italy, Sringer-Verlag.
- Kjeldskov, J. e Skov, M.B. (2003). Evaluating the Usability of Mobile Systems: Exploring Different Laboratory Approaches. In: *Proceedings of 10th International Conference on Computer-Human Interaction 2003 (CHI2003)*, Crete, Greece. Lawrence Erlbaum Associates.
- Kjeldskov, J., Skov, M.B., Als, B.S. e Høegh, R.T. (2004). Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the field. In: *Proceedings of the 6th International Mobile Conference (HCI 2004)*, Glasgow, Scotland. *Lecture Notes in Computer Science*, Berlin, Springer-Verlag.
- Koscianski, A. e Soares, M. S. (2007). *Qualidade de Software: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software*. 2 ed. São Paulo: Novatec, 2007. 395p.
- Kruchten, P. (2003). “Introdução ao RUP – Rational Unified Process. Tradução de Deborah Rüdiger. 2. ed. Rio de Janeiro: Ciência Moderna, 2003.
- Lee, V.; Schneider, H e Schell, R. (2005). *Aplicações Móveis - Arquitetura, projetos e desenvolvimento*. São Paulo: Pearson Education do Brasil, 2005.
- Lelli, V. L. e Andrade, R. M. C. (2008). Utilizando uma ferramenta de gerência de projetos para auxiliar no ensino de Engenharia de Software. In: *Fórum de Educação em Engenharia de Software - FEES 2008*, 2008, Campinas. Monografia em Ciência da Computação nº 43/ 08 - Fórum de Educação em Engenharia de Software. Rio de Janeiro: PUC - Pontifícia Universidade Católica do Rio de Janeiro, 2008. v. 43. p. 98-106.

- LioNBRIDGE (2009). LioNBRIDGE - Accelerating your global success. Disponível em: <<http://www.lionbridge.com/lionbridge.htm>>. Acesso em: 24 ago. 2009.
- LioNBRIDGE (2008). Shortening Mobile Device Testing Cycles: Best Practices for Testing Leadership in the Next Wave of Global Growth. Disponível em: <<http://www.lionbridge.com/lionbridge/en-US/kc/outsourced-testing.htm>>. Acesso em: 05 set. 2008.
- LioNBRIDGE (2006). Testing Mobile Applications is Different from Testing Traditional Applications. Disponível em: <<http://www.lionbridge.com/lionbridge/en-US/kc/outsourced-testing.htm>>. Acesso em: 05 set. 2008.
- LittleSpring (2009). Little Spring Design - Designing the mobile user experience. Disponível em: <<http://www.littlespringsdesign.com/>>. Acesso em: 24 ago. 2009.
- Mackenzie, I.S. e Sukoreff, R.W. (2002). Text entry for mobile computing: Models and methods, theory and practice. *Human-Computer Interaction*, 17, 147-198. 2002.
- Molinari, L. (2005). *Testes de Software: Produzindo Sistemas Melhores e Mais Confiáveis*. 2 ed. São Paulo: Érica, 2005.
- Motodev (2009). MOTODEV - The motorola developer network. Disponível em: <<http://developer.motorola.com/>>. Acesso em: 24 ago. 2009.
- Myers, J. G. (2004). *The Art of Software Testing*. Second edition Revised and updated by Tom Badgett and Todd M. Thomas with Corey Sandler. John Wiley and Sons, New Jersey, U.S.A., 2004. 234 p.
- Mikhalenko, P.V. (2006). *Best Practices for Mobile Web Application Development*. Tech Republic. Disponível em: <http://articles.techrepublic.com.com/5100-10878_11-6095452.html>. Acesso em: 9 set. 2008.
- Nguyen, H. Q. (2001). *Testing Applications on the Web, Test Planning for Internet-Based Systems*. USA: John Wiley & Sons, Inc., 2001. 402 p.
- Nielsen, Jacob. (1993). *Usability Engineering*. Boston - USA: Academic Press, 362 p.
- Nokia (2009). Forum Nokia - Driving mobile innovation. Disponível em: <<http://www.forum.nokia.com/>>. Acesso em: 24 ago. 2009.

- Nokia (2008). Nokia Test Criteria for Java™ ME Applications v1.1. Disponível em: <http://www.forum.nokia.com/main/technical_services/testing/testing_documents.htm>. Acesso em: 05 set. 2008.
- Nokia (2007). Testing Checklist for Symbian C++ Applications v1.2. Disponível em: <http://www.forum.nokia.com/main/technical_services/testing/testing_documents.htm>. Acesso em: 05 set. 2008.
- Nokia (2004). Usability, Culturally Speaking. Disponível em: <http://www.forum.nokia.com/main/resources/user_experience/usability/usability_articles.html>. Acesso em: 05 set. 2008.
- Nokia Forum (2009). Development and Quality Assurance Process for Mobile Applications. Disponível em: <http://www.forum.nokia.com/Technology_Topics/Application_Quality/Quality/>. Acesso em: 03 ago. 2009.
- NokiaResearch (2009). Nokia Researcher Center. Disponível em: <<http://research.nokia.com/research/index.html>>. Acesso em: 24 ago. 2009.
- Overlander, D. (2008). Usability and the Design of Mobile Interfaces. Disponível em: <<http://www.pocketpcmag.com/cms/node/1346>>. Acesso em 05 set. 2008.
- Pocatilu, P. (2008). Testing Java Me applications. In: Revista Informática Econômica, vol. 3, n. 47. 2008.
- Pressman, R. S. (2006). Engenharia de Software. Makron Books, 6 ed. 2006.
- Roto, V.; Oulasvirta, A. Haikarainen; T. Lehmuskallio, H. & Nyyssönen, T. (2004). Examining mobile phone use in the wild with uasi-experimentation. HIIT Technical Report 2004-1, August 13.
- Roto, V.; Ketola, P. e Huotari, S. (2008). User Experience Evaluation in Nokia. Now Let's Do It in Practice - User Experience Evaluation Methods in Product Development workshop in Conference on Human Factors in Computing Systems archive (CHI'08), Florence, Italy, 2008.

- Santos-Neto, P. A. (2006) MODEST: Um Método de Teste Baseado em Modelos. Belo Horizonte: UFMG, 2004. 138p Tese de Doutorado, Doutorado em Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte, 2006.
- Serco (2009). Serco - bringing service to life Disponível em <<http://www.serco.com/usability/>>. Acesso em: 24 ago. 2009.
- Silveira, M.; Pinho, M.; Gonella, A. e Calvetti, P. (2004). Avaliando a Qualidade da Interação de Usuários com Dispositivos Móveis. In: Anais do VI Simpósio sobre Fatores Humanos em Sistemas Computacionais – Mediando e Transformando o Cotidiano, Curitiba, Brasil, 2004, p. 37-46.
- Smartphonemag (2009). Smartphonemag – Smartphone & Pocket PC magazine. Disponível em: <<http://www.smartphonemag.com/>>. Acesso em: 24 ago. 2009.
- SonyEricsson (2009). Sony Ericsson - Developer World. Disponível em: <http://developer.sonyericsson.com/site/global/home/p_home.jsp>. Acesso em: 24 ago. 2009.
- Sony Ericsson developer (2009). Java ME Platform Docs & Tools. Disponível em: <http://developer.sonyericsson.com/site/global/docstools/java/p_java.jsp>. Acesso em: 20 ago. 2009.
- Sun microsystems (2009). Sun Java Wireless Toolkit 2.5.2_01 for CLDC. Disponível em: <<http://java.sun.com/products/sjwtoolkit/download.html>>. Acesso em: 20 ago. 2009.
- Rubin, J. (1994). Handbook of Usability Testing: how to plan, design and conduct effective tests. USA: John Wiley & Sons, Inc. 1994, 330 p.
- Sommerville, I. (2007). Engenharia de Software. Addison Wesley, 8 ed. 2007.
- TechRepublic (2009). TechRepublic. Disponível em: <<http://techrepublic.com/>>. Acesso em 24 ago. 2009.
- TestQuest Pro (2004) “Automating Functional Testing of Mobile Business Application Software for a Competitive Advantage”, TESTQUEST, INC., Minneapolis, MN 55317, USA, 2004. Disponível em: <<http://testquest.com/download.cfm?oid=4358&product=3487>>. Acesso em 22 dez. 2008.

- TestQuest (2009). Test Quest by Square. Disponível em:< <http://testquest.com/mobile.cfm>>. Acesso em: 24 ago. 2009.
- Weiss, S. (2002). Handheld Usability. John Wiley & Sons, Ltd, West Sussex, England, 2002.
- WedCredible (2009). WedCredible User experience research & design. Disponível em: <<http://www.webcredible.co.uk/>>. Acesso em: 24 ago. 2009.
- W3C (2009). W3C - World Wide Web. Disponível em <<http://www.w3.org/>>. Acesso em: 24 ago. 2009.
- Zeidler, C.; Kittl, C. e Petrovic, O. (2007). An Integrated Product Development Process for Mobile Software. In: Proceedings of the IX International Conference on the Management of Mobile Business (ICBM), 2007, p. 23-23.
- Zhang, D. and Adipat, B. (2005) “Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications”. In: Proceedings of the International Journal of Human Computer Interaction (IJHCI), vol. 18, n° 3, 2005, p. 293-308.

Anexo A – Casos de testes das aplicações móveis

Este anexo apresenta os casos de testes das aplicações Shopping Móvel, Museu Móvel e Teatro Móvel.

Shopping Móvel

Procedimentos de teste da aplicação cliente:

Identificação	C01
Descrição	Visualizar lojas
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	-
Ação a ser tomada	Selecionar a ação “Visualizar Lojas” na tela inicial.
Resultado esperado	O sistema deve apresentar uma lista de categorias de lojas.
Método para validação	Apresentação da lista de categorias de lojas.

Identificação	C02
Descrição	Visualizar lojas de uma categoria
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	O sistema se encontra na lista de categorias.
Ação a ser tomada	Selecionar uma das categorias apresentadas.
Resultado esperado	O sistema deve apresentar uma lista de lojas da categoria selecionada.
Método para validação	Apresentação da lista de lojas da categoria selecionada

Identificação	C03
Descrição	Visualizar produtos de uma loja.
Pré-condições	O sistema se encontra na tela de uma loja.
Condições de entrada	Pelo menos um produto da loja está cadastrado.
Ação a ser tomada	Selecionar uma categoria apresentada e em seguida selecionar um produto apresentado.
Resultado esperado	O sistema deve apresentar uma lista de categorias de produtos que a loja dispõe e em seguida apresentar as características do produto escolhido, fotos, preço, quantidade economizada se o produto se encontra em promoção.
Método para validação	Apresentação das características do produto escolhido.

Identificação	C04
Descrição	Comprar Produto em uma loja sem ter feito <i>login</i> .
Pré-condições	O sistema se encontra na tela de um produto sem o usuário ter feito <i>login</i> .
Condições de entrada	Pelo menos um produto da loja está cadastrado.
Ação a ser tomada	Pressionar o botão de compra.
Resultado esperado	O sistema deve levar o usuário para a tela de <i>login</i> .
Método para validação	O sistema não deve deixar um usuário que não esteja logado comprar o produto.

Identificação	C05
Descrição	Comprar Produto em uma loja tendo feito <i>login</i> .
Pré-condições	O sistema se encontra na tela de uma loja e o usuário já deve ter feito <i>login</i> com sucesso.
Condições de entrada	Pelo menos um produto da loja e um usuário estão cadastrados.
Ação a ser tomada	Pressionar o botão comprar, preencher o formulário e tentar efetuar a compra.
Resultado esperado	O sistema deve tanto apresentar uma tela com a frase compra realizada com sucesso ou uma mensagem avisando quanto de estoque a loja possui no momento.
Método para validação	Efetivar a compra se o estoque for maior que a quantidade pedida, não efetivar caso contrário.

Identificação	C06
Descrição	Visualizar grupos de produtos.
Pré-condições	O sistema se encontra na tela de uma loja.
Condições de entrada	Pelo menos um grupo de produtos foi cadastrado.
Ação a ser tomada	Solicitar a lista de produtos.
Resultado esperado	O sistema deve apresentar a lista de categorias cadastradas.
Método para validação	O sistema apresenta a lista de categorias cadastradas.

Identificação	C07
Descrição	Visualizar promoções não estando cadastrado.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	Pelo menos um produto em promoção foi cadastrado.
Ação a ser tomada	Solicitar a lista de promoções
Resultado esperado	O sistema deve apresentar todos os produtos que estão em promoção.
Método para validação	O sistema apresenta a lista com todos os produtos que estão em promoção

Identificação	C08
Descrição	Visualizar promoções estando logado.
Pré-condições	O sistema se encontra na tela inicial, após ter sido realizado um <i>login</i> .
Condições de entrada	Pelo menos um produto em promoção e um usuário foram cadastrados.
Ação a ser tomada	Solicitar a lista de promoções.
Resultado esperado	O sistema deve apresentar os produtos que estão em promoção e que fazem parte das preferências do usuário.
Método para validação	O sistema apresenta a lista com os produtos que estão em promoção, mas somente aqueles que fazem parte das preferências do usuário.

Identificação	C09
Descrição	Cadastro de cliente.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	-
Ação a ser tomada	Ir para a tela de fazer <i>login</i> , em seguida escolher cadastrar e finalmente preencher o formulário e submeter ao servidor.
Resultado esperado	O sistema deve avisar que o cadastro foi um sucesso ou avisar o usuário se algum campo foi deixado em branco.
Método para validação	O sistema faz o cadastro com sucesso quando todas as informações são fornecidas.

Identificação	C10
Descrição	Cadastro de cliente, com <i>login</i> já cadastrado.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	Pelo menos um usuário já deve ter sido cadastrado
Ação a ser tomada	Ir para a tela de fazer <i>login</i> , em seguida escolher cadastrar e finalmente preencher o formulário com o <i>login</i> de um usuário já cadastrado e submeter ao servidor.
Resultado esperado	O sistema deve avisar que o <i>login</i> desejado já está em uso.
Método para validação	O sistema não permite que o usuário se cadastre.

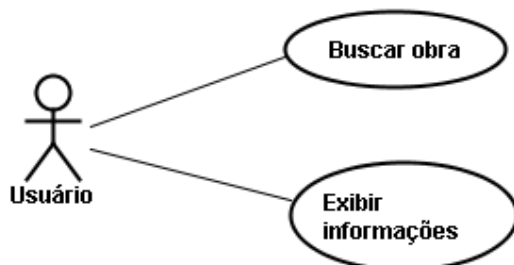
Identificação	C11
Descrição	Cadastro de cliente, com e-mail já cadastrado.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	Pelo menos um usuário já deve ter sido cadastrado
Ação a ser tomada	Ir para a tela de fazer <i>login</i> , em seguida escolher cadastrar e finalmente preencher o formulário com o e-mail de um usuário já cadastrado e submeter ao servidor.
Resultado esperado	O sistema deve avisar que o e-mail desejado já está em uso.
Método para validação	O sistema não permite que o usuário se cadastre.

Identificação	C12
Descrição	<i>Login</i> com usuário não cadastrado.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	Pelo menos um usuário já deve ter sido cadastrado.
Ação a ser tomada	Ir para a tela de fazer <i>login</i> e tentar logar com um <i>login</i> não cadastrado.
Resultado esperado	O sistema deve avisar que o usuário não está cadastrado.
Método para validação	O sistema não permite que o usuário se logue.

Identificação	C13
Descrição	<i>Login</i> usando uma senha errada.
Pré-condições	O sistema se encontra na tela inicial.
Condições de entrada	Pelo menos um usuário já deve ter sido cadastrado.
Ação a ser tomada	Ir para a tela de fazer <i>login</i> e tentar logar com uma senha incorreta.
Resultado esperado	O sistema deve avisar que a senha está incorreta.
Método para validação	O sistema não permite que o usuário se logue.

Museu Móvel

Procedimentos de teste da aplicação cliente:

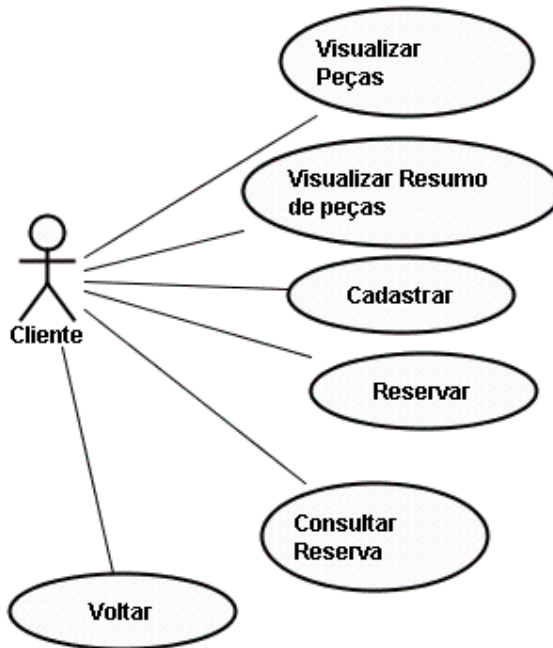


O diagrama de caso de teste envolve a busca e a exibição de informações de uma obra. A busca deverá ser efetuada por título de obra, nome de autor ou exposição. Na exibição das informações da obra, o sistema deverá informar: título da obra, nome do autor, ano de criação, obra permanente, exposição, imagem, informações básicas e informações extras.

Procedimento de Teste	Pré-condições	Condições de Entrada	Ação a ser Tomada	Resultado Esperado	Método para Validação
Buscar por autor da obra	Autor previamente incluído no banco de dados	Informar o nome do autor	Executar a thread Buscar	Uma lista com as obras do autor informado.	Dados listados (títulos das obras) na tela.
Título da obra	Obra previamente incluída no banco de dados	Informar o título da obra	Executar a thread Buscar	Uma lista com as obras cujo título seja (total ou em parte) o informado	Dados listados (títulos das obras) na tela
Buscar por exposição	Exposição previamente incluída no banco de dados	Informar a exposição à qual a obra está ligada	Executar a thread Buscar	Uma lista com as obras pertencentes àquela exposição	Dados listados (títulos das obras) na tela
Buscar por autor/título/exposição inválido ou inexistente.	-	Informar um nome de autor/título de obra/exposição inválido ou inexistente	Executar a thread Buscar	Voltar para a tela de busca para nova entrada de dados	Apresentação da mensagem: "Autor/Título/Exposição Inválido ou Inexistente".
Exibir Informações	Busca efetuada	Selecionar a obra desejada	Executar a thread MostrarInfo	Informações da obra desejada na tela	Informações da obra

Teatro Móvel

Procedimentos de teste da aplicação cliente:



Na visualização de Peças o sistema deverá informar: nomes das peças;
 Na visualização de Resumo de Peças o sistema deverá informar: resumo das peças;
 Na visualização de reservas o sistema deverá informar: números seriais;
 Na reserva o usuário deve fornecer: *login*, senha, quantidade;
 Na consulta de reserva o usuário deve fornecer: *login* e senha.

Procedimento de Teste	Pré-condições	Condições de Entrada	Ação a ser Tomada	Resultado Esperado	Método para Validação
Cadastrar usuário	-	Informar dados do usuário	Executar InsereCadastro	Dados incluídos no MySql	Mensagem de Confirmação
Reservar Bilhete	<i>Login</i> e Senha confirmados	Peça e quantidade	Executar InsereReserva	Reserva Incluída no banco de dados	Mensagem de Confirmação

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.