



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

**Dissertação de Mestrado**

**Características e Medidas de *Software* para Avaliação da Qualidade  
da Interação Humano-Computador em Sistemas Ubíquos**

**Rainara Maia Santos**

Fortaleza – Ceará  
2014

**Rainara Maia Santos**

**Características e Medidas de *Software* para Avaliação da Qualidade da Interação Humano-Computador em Sistemas Ubíquos**

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.  
Co-orientadora: Káthia Marçal de Oliveira, Dra.

Fortaleza – Ceará  
2014

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca de Ciências e Tecnologia

- 
- S238c Santos, Rainara Maia.  
Características e medidas de software para avaliação da qualidade da interação humano-computador em sistemas ubíquos / Rainara Maia Santos. – 2014.  
166 f. : il. color.
- Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Departamento de Computação, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2014.  
Área de Concentração: Ciência da Computação.  
Orientação: Profa. Dra. Rossana Maria de Castro Andrade.  
Coorientação: Profa. Dra. Káthia Marçal de Oliveira.
1. Computação Ubíqua. 2. Interação homem-máquina. 3. Software - Controle de qualidade. I.  
Título.

---

CDD 005


RAINARA MAIA SANTOS

**Características e Medidas de *Software* para Avaliação da Qualidade da Interação Humano-Computador em Sistemas Ubíquos**

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 28 de fevereiro de 2014


BANCA EXAMINADORA

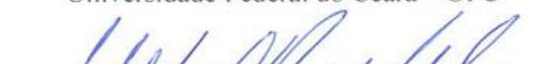
  
\_\_\_\_\_  
Prof. Dra. Rossana Maria de Castro Andrade  
(Orientadora)  
Universidade Federal do Ceará – UFC

  
\_\_\_\_\_  
Prof. Dra. Káthia Marçal de Oliveira  
(Coorientadora)  
Universidade de Valenciennes et do Hainaute-Cambrésis

  
\_\_\_\_\_  
Prof. Dr. Adriano Bessa Albuquerque  
Universidade de Fortaleza – Unifor

  
\_\_\_\_\_  
Prof. Dra. Andréia Libório Sampaio  
Universidade Federal do Ceará – UFC

  
\_\_\_\_\_  
Prof. Dr. Lincoln Souza Rocha  
Universidade Federal do Ceará – UFC

  
\_\_\_\_\_  
Prof. Dr. Miguel Franklin de Castro  
Universidade Federal do Ceará – UFC

Fortaleza, 28 de fevereiro de 2014.

Dedico essa dissertação à Maria Venilce Maia,  
minha avó.

# Agradecimentos

Agradeço primeiramente a minha avó Venilce, um anjo da guarda que Deus colocou na minha vida. Obrigada Vó, por todas as orações diárias concedidas a mim. Eu espero ser um dia a mulher que a senhora é.

Agradeço a Deus por atender as orações da minha avó, por me dar forças em momentos difíceis e, principalmente, por ter me agraciado com uma vida cheia de oportunidades.

Aos meus queridos Pais, por todos os seus esforços para que não me faltasse nada, por serem grandes incentivadores e por sempre confiarem nas minhas decisões.

Aos meus irmãos por trazerem muito amor na minha vida. A minha primeira sobrinha e afilhada, Ingrid, por despertar o que há de melhor em mim.

A todos os meus familiares: avôs, padrinhos, tios e primos, que sempre me trataram como filha e como irmã. Todos me ajudaram muito ao longo da minha vida acadêmica. Sou muito feliz por fazer parte de uma família como a minha.

Ao meu noivo Jefferson, pois eu sei que essa também é uma conquista dele. Assim como as dele são minhas. Espero que essa seja uma de muitas que estão por vir em nossas vidas. Aproveito e agradeço à família dele também que, com muito carinho, me acolheram e me ajudaram em diversos momentos.

Agradeço, especialmente, as minhas orientadoras, Rossana e Káthia, por terem me concedido tantas oportunidades maravilhosas ao longo do mestrado, por acreditarem em mim, por me ensinarem, por me inspirarem. Que venham mais quatro anos de parceria.

Aos meus amigos do GREat, em especial, os colegas do MDCC, da CTQS e do projeto Maximum. Obrigada por todas as discussões, pelas revisões e pelos momentos de diversão compartilhados durante esses dois anos. Foi um grande prazer.

A todos os funcionários do GREat e MDCC, especialmente: Orley, Marquinhos, Darilú, Janaína e Géssika. Aos professores das disciplinas que fiz durante o mestrado e aos professores da UFC em Quixadá, campus onde “tudo” começou.

Agradeço a CAPES e a FUNCAP pelo suporte financeiro, o qual foi imprescindível para a realização deste trabalho.

Por último, agradeço as pessoas que, com muita boa vontade, participaram das avaliações das aplicações.

“De tudo, ficaram três coisas: a certeza de que ele estava sempre começando, a certeza de que era preciso continuar e a certeza de que seria interrompido antes de terminar. Fazer da interrupção um caminho novo. Fazer da queda um passo de dança, do medo uma escada, do sono uma ponte, da procura um encontro.”

Fernando Sabino

# Resumo

Os sistemas ubíquos modificam a forma como os usuários interagem com os computadores, pois seus serviços devem estar em toda parte, auxiliando-os em diversas atividades do cotidiano. Esse auxílio deve acontecer de forma transparente, no qual nenhuma ou pouca atenção e entrada de dados do usuário são necessárias. Avaliar a qualidade desse tipo de sistema se torna então um elemento essencial para garantir sua adoção. No entanto, as características específicas de sistemas ubíquos, como a sensibilidade ao contexto e o novo tipo de interação, que é a interação implícita e transparente, sugerem que novas medidas de *software* sejam levadas em consideração na avaliação da qualidade desses sistemas. O objetivo desse trabalho é, portanto, definir medidas de *software* para a avaliação da qualidade da interação humano-computador em sistemas ubíquos. Para tanto, primeiro, são identificadas as características de qualidade essenciais dos sistemas ubíquos que afetam a interação do usuário. Tal identificação é realizada através de um mapeamento sistemático que visa coletar o que já está sendo definido e avaliado na literatura. Após isso, o método *Goal-Question-Metric* (GQM) é utilizado para definir as medidas de *software* deste trabalho. O conjunto de medidas propostas é então aplicado através de estudos de caso envolvendo três aplicações ubíquas desenvolvidas para dispositivos móveis.

**Palavras-chave:** Sistemas Ubíquos, Interação Humano-Computador e Qualidade de Software.



# Abstract

Ubiquitous systems change the way users interact with computers, because their services must be everywhere, supporting users in several everyday activities in a transparent way with little or no need for attention or data input from a user. So, an essential element to ensure these systems adoption is to assess their quality. However, the ubiquitous systems specific characteristics, like context-awareness and the new type of interaction, that is the implicit and transparent interaction, suggest taking into account new software measures in these systems's quality assessment. Thus, the aim of this work is to define software measures to human-computer interaction quality assessment in ubiquitous systems. To achieve this goal, first it is identified essential quality characteristics of ubiquitous systems that impact on user interaction. Such identification is done through a systematic mapping, to collect what has been already defined and evaluated in the literature. After that, the Goal-Question-Metric (GQM) method is used to define the software measures proposed in this work. The set of proposed measures is evaluated through case studies with three ubiquitous applications developed to mobile devices.

**Key-words:** Ubiquitous Systems, Human-Computer Interaction and Software Quality.

# Lista de Figuras

Figura 1.1 – Metodologia de Trabalho.....	21
Figura 2.1 - Modelo de contexto. Fonte: (BACHA; OLIVEIRA; ABED, 2011).....	28
Figura 2.2 – Camadas de Abstração e Interpretação de Contexto, traduzida de (BETTINI et al., 2010) ...	29
Figura 2.3 - Arquitetura Aura, traduzida de (GARLAN; SIEWIOREK; STEENKISTE, 2002).....	31
Figura 2.4 - Arquitetura SysSu. Fonte: (LIMA et al., 2011).....	32
Figura 2.5 - Arquitetura do Mobiline. Fonte: (MARINHO et al., 2010).....	33
Figura 2.6 - Exemplo da Escala de Likert e Diferencial Semântico. Fonte: (BARBOSA; SILVA, 2010) ...	37
Figura 2.7 - Divisão da norma SQuaRE, traduzida de (ISO/IEC 25000, 2005) .....	44
Figura 2.8 - Estrutura de um Modelo de Qualidade, traduzida de (ISO/IEC 25000, 2005) .....	45
Figura 2.9 - Modelo de Qualidade da ISO/IEC 9126, traduzida de (MACHADO; SOUZA, 2004).....	45
Figura 2.10 - Relacionamento entre EMQ, função de medição, medida de qualidade e modelo de qualidade, traduzida de (ISO/IEC 25000, 2005).....	49
Figura 2.11 - Exemplo de elementos de qualidade, função de medição e medida de qualidade, traduzida de (ISO/IEC 25000, 2005).....	49
Figura 2.12 - Fases do Paradigma GQM, traduzida de (VAN SOLINGEN; BERGHOUT, 1999).....	52
Figura 2.13 - Estrutura do Paradigma GQM, traduzida de (VAN SOLINGEN; BERGHOUT, 1999).....	52
Figura 2.14 - <i>Abstraction Sheets</i> , traduzida de (VAN SOLINGEN; BERGHOUT, 1999) .....	53
Figura 3.1 - Processo de MS, adaptado de (NETO et al., 2011)(PETERSEN et al., 2008).....	58
Figura 3.2 – Quantidade de Estudos vs Fonte de Pesquisa.....	61
Figura 3.3 - Filtros .....	61
Figura 3.4 - Ano de Publicação .....	62
Figura 3.5 - Distribuição de Artigos por Tipo de Publicação .....	65
Figura 3.6 - Domínios de Aplicações Ubíquas .....	65
Figura 3.7 - Número de Características Extraídas por Estudo .....	67
Figura 3.8 – Número de Medidas de <i>Software</i> Extraídas por Estudo .....	68
Figura 3.9 - Estrutura de Avaliação da Interatividade, traduzida de (LEE et al., 2008).....	69
Figura 3.10 - Distribuição por tipo de contribuição .....	71
Figura 3.11 - Distribuição por tipo de pesquisa .....	71
Figura 3.12 - O Mapa Sistemático .....	71
Figura 4.1 – Fluxo de Atividades das Entrevistas GQM.....	84
Figura 4.2 – <i>Abstract Sheet</i> para Sensibilidade ao Contexto .....	85
<b>Figura 4.3 - <i>Abstract Sheet</i> para Mobilidade.....</b>	<b>87</b>
Figura 4.4 - <i>Abstract Sheet</i> para Transparência .....	88
Figura 4.5 - <i>Abstract Sheet</i> para Foco.....	89

Figura 4.6- <i>Abstract Sheet</i> para Interação Calma .....	90
Figura 4.7 – Resumo dos objetivos de medição, questões e medidas.....	109
Figura 5.1 – Aplicação UbiPrinter .....	114
Figura 5.2 – Aplicação UbiMute .....	122
Figura 5.3 - Aplicação GREat Tour, adaptada de (LIMA et al., 2013) .....	129

# Lista de Tabelas

Tabela 2.1 - Medidas para Subcaracterística Compreensibilidade, traduzida de (ISO/IEC 9126, 2001).....	50
Tabela 2.2 - Medidas para a Subcaracterística Estabilidade, traduzida de (ISO/IEC 9126, 2001) .....	51
Tabela 2.3 - Template para Definir Objetivos, traduzida de (VAN SOLINGEN; BERGHOUT, 1999) ..	53
Tabela 3.1 - Termos chave.....	59
Tabela 3.2 - Sumário de artigos selecionados por fonte e filtro.....	62
Tabela 3.3 - Faceta Tipo de Contribuição .....	63
Tabela 3.4 - Faceta Tipo de Pesquisa, traduzida de (PETERSEN et al., 2008) .....	64
Tabela 3.5 - Formulário de Extração de Dados .....	64
Tabela 3.6. Exemplo de Características Extraídas.....	66
Tabela 3.7 - <i>Framework</i> de Avaliação de Sistemas Ubíquos, traduzida de (SCHOLTZ; CONSOLVO, 2004) .....	70
Tabela 3.8 – Lista de Características de Qualidade da IHC em Sistemas Ubíquos .....	73
Tabela 4.1 – Objetivo de Medição.....	83
Tabela 4.2 – Medidas de Software para a Questão 1 .....	94
Tabela 4.3 - Medidas de Software para a Questão 2.....	96
Tabela 4.4 – Medidas de Software para a Questão 3.....	98
Tabela 4.5 –Medidas de software para a Questão 4.....	101
Tabela 4.6 – Medidas de Software para a Questão 5.....	104
Tabela 4.7 - Medidas de software para a Questão 6 .....	106
Tabela 5.2 – Procedimento para Realização da Avaliação do UbiPrinter .....	115
Tabela 5.3 – Resultados da UbiPrinter para a Questão 1.....	115
Tabela 5.4 - Resultados da UbiPrinter – Questão 2.....	117
Tabela 5.5 - Resultados da UbiPrinter – Questão 3.....	117
Tabela 5.6 - Resultados UbiPrinter – Questão 4.....	118
Tabela 5.7 - Resultados UbiPrinter – Questão 5.....	120
Tabela 5.8 - Resultados UbiPrinter – Questão 6.....	121
Tabela 5.9 - Resultados da UbiMute para a Questão 1 .....	123
Tabela 5.10 - Resultados da UbiMute para a Questão 2 .....	124
Tabela 5.11 - Resultados da UbiMute para a Questão 3 .....	125
Tabela 5.12 - Resultados da UbiMute para a Questão 4.....	126
Tabela 5.13 - Resultados UbiMute– Questão 5.....	127
Tabela 5.14 - Resultados UbiMute – Questão 6.....	128
Tabela 5.15 - Procedimento para Realização da Avaliação do GREat Tour .....	130
Tabela 5.16 – Resultados GREat Tour – Questão 1 .....	131

Tabela 5.17 – Resultados GREat Tour – Questão 2.....	131
Tabela 5.18 – Resultados GREat Tour – Questão 3.....	132
Tabela 5.19 – Resultados GREat Tour – Questão 4.....	133
Tabela 5.20 - Resultados GREat Tour–Questão 5.....	134
Tabela 5.21 – Resultados GREat Tour – Questão 6.....	135
Tabela 0.1 – Medidas de Software Encontradas pelo Mapeamento Sistemático.....	150
Tabela 0.2 - Classificação dos Estudos do Mapeamento Sistemático.....	160

# Lista de Abreviaturas

IHC	Interação Humano-Computador
MS	Mapeamento Sistemático
RS	Revisão Sistemática
QP	Questão de Pesquisa
ISO	<i>International Organisation for Standardisation</i>
GQM	<i>Goal-Question-Metric</i>
EC	Estudo de Caso
GPS	<i>Global Positioning System</i>
QRCode	<i>Quick Response Code</i>
GREat	Grupo de Redes de Computadores, Engenharia de <i>Software</i> e Sistema
GPRS	<i>General Packet Radio Service</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
HSPA	<i>High Speed Packet Access</i>
LTE	<i>Long Term Evolution</i>
WiMAX	<i>Worldwide Interoperability for Microwave Access</i>
DVB-H	<i>Digital Video Broadcasting - Handheld</i>
SO	Sistema Operacional

# Sumário

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>17</b>
1.1	CONTEXTUALIZAÇÃO.....	17
1.2	MOTIVAÇÃO.....	19
1.3	OBJETIVOS.....	20
1.4	METODOLOGIA.....	20
1.5	ORGANIZAÇÃO DA DISSERTAÇÃO.....	21
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>23</b>
2.1	SISTEMAS UBÍQUOS.....	23
2.1.1	<i>Definições e Exemplos</i> .....	23
2.1.2	<i>Características</i> .....	25
2.1.3	<i>Sensibilidade ao Contexto</i> .....	26
2.1.4	<i>Arquiteturas</i> .....	29
2.1.5	<i>Desafios de Pesquisa</i> .....	33
2.2	INTERAÇÃO HUMANO-COMPUTADOR.....	35
2.2.1	<i>Definições</i> .....	35
2.2.2	<i>Métodos de Avaliação</i> .....	37
2.2.3	<i>IHC em Sistemas Ubíquos</i> .....	40
2.2.4	<i>Dificuldades na Avaliação da IHC em Sistemas Ubíquos</i> .....	41
2.3	QUALIDADE DE SOFTWARE.....	42
2.3.1	<i>Definições</i> .....	42
2.3.2	<i>A Norma SQuaRE</i> .....	43
2.3.3	<i>Medição e Medidas de Software</i> .....	47
2.3.4	<i>Definição de Medidas nas Normas</i> .....	48
2.3.5	<i>Abordagem para Definição de Medidas</i> .....	51
2.4	CONSIDERAÇÕES FINAIS.....	54
<b>3</b>	<b>INVESTIGAÇÃO E DEFINIÇÃO DE CARACTERÍSTICAS DE QUALIDADE PARA IHC EM SISTEMAS UBÍQUOS.....</b>	<b>56</b>
3.1	METODOLOGIA DE BUSCA: MAPEAMENTO SISTEMÁTICO.....	56
3.2	PLANEJAMENTO.....	58
3.3	EXECUÇÃO.....	60
3.3.1	<i>Condução da Busca</i> .....	60
3.3.2	<i>Filtragem</i> .....	61
3.3.3	<i>Keywording de Tópicos Relevantes e Extração de Dados</i> .....	63
3.4	RESULTADOS.....	65
3.4.1	<i>Resposta das Questões de Pesquisa</i> .....	66
3.4.2	<i>O Mapa Sistemático</i> .....	70
3.5	DEFINIÇÃO DO CONJUNTO FINAL DE CARACTERÍSTICAS DE QUALIDADE.....	73
3.6	CONSIDERAÇÕES FINAIS.....	78
<b>4</b>	<b>MEDIDAS DE SOFTWARE PARA AVALIAÇÃO DA IHC EM SISTEMAS UBÍQUOS.....</b>	<b>80</b>
4.1	SELEÇÃO DE CARACTERÍSTICAS ESPECÍFICAS DE IHC PARA SISTEMAS UBÍQUOS.....	80
4.2	PLANEJAMENTO PARA ELABORAÇÃO DAS MEDIDAS.....	82
4.3	DEFINIÇÃO DAS MEDIDAS.....	83
4.3.1	<i>Objetivo de Medição</i> .....	83
4.3.2	<i>Entrevistas GQM</i> .....	84
4.3.3	<i>Abstract Sheets</i> .....	85
4.3.4	<i>Questões</i> .....	91
4.3.5	<i>Medidas</i> .....	92
4.4	COLETA DE DADOS.....	110
4.5	INTERPRETAÇÃO DOS RESULTADOS.....	110
4.6	CONSIDERAÇÕES FINAIS.....	111

<b>5 ESTUDO DE CASO .....</b>	<b>112</b>
5.1 PLANEJAMENTO .....	112
5.2 UBIPRINTER .....	113
5.2.1 <i>Coleta e Interpretação dos Resultados</i> .....	115
5.3 UBI MUTE.....	121
5.3.1 <i>Coleta e Interpretação dos Resultados</i> .....	122
5.4 GREAT TOUR.....	128
5.4.1 <i>Coleta e Interpretação dos Resultados</i> .....	130
5.5 DISCUSSÃO DOS RESULTADOS .....	136
5.6 CONSIDERAÇÕES FINAIS .....	137
<b>6 CONCLUSÃO.....</b>	<b>139</b>
6.1 RESULTADOS ALCANÇADOS .....	139
6.2 PRODUÇÃO BIBLIOGRÁFICA.....	140
6.3 TRABALHOS FUTUROS .....	141
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>143</b>
<b>APÊNDICE A – MEDIDAS DE SOFTWARE .....</b>	<b>150</b>
<b>APÊNDICE B – CLASSIFICAÇÃO DOS ESTUDOS .....</b>	<b>160</b>
<b>APÊNDICE C – FORMULÁRIO PARA DEFINIÇÃO DO PERFIL DO USUÁRIO .....</b>	<b>162</b>
<b>APÊNDICE D – FORMULÁRIO PARA COLETA DE DADOS APÓS O USO DA APLICAÇÃO .....</b>	<b>163</b>
<b>APÊNDICE E – FORMULÁRIO DE AVALIAÇÃO PARA O DESENVOLVEDOR.....</b>	<b>165</b>
<b>APÊNDICE F – FORMULÁRIO DE OBSERVAÇÃO PARA O AVALIADOR .....</b>	<b>166</b>



# 1 Introdução

Essa dissertação tem como objetivo propor um conjunto de medidas de *software* para avaliação de características específicas e essenciais de interação humano-computador em sistemas ubíquos. Este capítulo introduz, na Seção 1.1, o contexto em que o trabalho de dissertação está inserido. A Seção 1.2 apresenta a motivação para o desenvolvimento do trabalho. A Seção 1.3 apresenta os objetivos da dissertação. A Seção 1.4 apresenta a metodologia utilizada para realização desse trabalho e, por fim, a Seção 1.5 apresenta a estrutura organizacional dessa dissertação.

## 1.1 Contextualização

Ao longo das últimas décadas de computação, duas grandes tendências caracterizaram o relacionamento entre o usuário e a tecnologia (WEISER; BROWN, 1997). A primeira é chamada de “*mainframe*” e é caracterizada pelo relacionamento “um computador, vários usuários”. A segunda é a era dos computadores pessoais, que é caracterizada pelo relacionamento “um computador, um usuário”.

Uma terceira tendência caracterizada pelo relacionamento “um usuário, muitos computadores” surgiu com o avanço da internet, da computação distribuída e da computação móvel (ROCHA, 2007). Esta tendência de um usuário e vários computadores está diretamente relacionada ao paradigma idealizado por Mark Weiser, denominado de Computação Ubíqua (WEISER, 1991). Weiser (1991) descreve sua visão de computação ubíqua através da seguinte frase: “As tecnologias mais profundas são aquelas que desaparecem. Elas se misturam com os objetos do dia-a-dia até se tornarem indistinguíveis”.

A principal motivação por trás da visão da computação ubíqua é auxiliar as atividades diárias das pessoas e mudar completamente a experiência de uso com os computadores (ABOWD; MYNATT; RODDEN, 2002). No ambiente ubíquo, os sistemas estão inseridos nos objetos do dia-a-dia e são capazes de monitorar o usuário e seu ambiente a fim de fornecer serviços relevantes, sem que o usuário perceba que está interagindo com uma infraestrutura computacional. Assim, a interação com esse tipo de sistema deve ser invisível ou com um mínimo de distração por parte do usuário.

Os avanços nos dispositivos computacionais, como a capacidade de embutir sensores para capturar informações do usuário e seu ambiente, e nos meios de comunicação sem fio têm tornando real o desenvolvimento de sistemas ubíquos.

Tais sistemas precisam ser entregues para o usuário priorizando a qualidade da interação, pois no domínio da computação ubíqua, é muito mais preocupante a seguinte questão: “Quem quer ter centenas de computadores ao redor requisitando constantemente atenção e nos bombardeando com informações irrelevantes?” (RIEKKI; ISOMURSU; ISOMURSU, 2004).

Como os sistemas ubíquos se propõem a facilitar diversas tarefas simples do cotidiano a qualquer hora e em qualquer lugar, um sistema desenvolvido que não auxilie corretamente os usuários nas suas tarefas do dia-a-dia causando perturbação, desconforto e insegurança, certamente desmotiva um usuário para utilizá-lo.

Para determinar a qualidade de produtos de *software*, incluindo a qualidade da interação, medições de *software* podem ser realizadas. O propósito de realizar medições é obter dados caracterizando um produto com relação a uma série de características que se deseja que o *software* alcance. Por exemplo: *Usabilidade, Manutenibilidade e Confiabilidade* são exemplos de características que um *software* pode possuir. (KOSCIANSKI; SOARES, 2006).

Essas características geralmente estão organizadas em um modelo, que é chamado de Modelo de Qualidade pelas normas (ISO/IEC 9126, 2001) (ISO/IEC 25000, 2005). Ele é formado por uma definição hierárquica de características de qualidade e medidas de *software* que permitam a avaliação de um produto.

Um dos mais conhecidos modelos é o definido pela norma (ISO/IEC 9126, 2001), atualmente incorporada na SQuaRE (*Software Product Quality Requirements and Evaluation*) (ISO/IEC 25000, 2005). Esse modelo de qualidade é para sistemas em geral, não possuindo características ou medidas de *software* inerentes aos sistemas ubíquos ou outras tecnologias específicas.

Em relação à Interação Humano-Computador (IHC), a norma especifica a característica de qualidade *Usabilidade* e medidas de *software* para avaliá-la. Essa característica representa a capacidade do *software* de ser entendido, aprendido, usado e atrativo para o usuário, quando usado sob condições específicas (ISO/IEC 25000, 2005). Por outro lado, a norma (ISO 9241-11, 1998) define *Usabilidade* como sendo: “a eficiência, a eficácia e a satisfação com as quais determinados usuários realizam determinadas atividades em um contexto de uso”.

No entanto, características específicas de sistemas ubíquos, por exemplo, a sensibilidade ao contexto, e um novo tipo de interação, que é a interação implícita e transparente com o usuário sugerem que novas características e medidas de *software* devem ser levadas em consideração em avaliações da qualidade da IHC de sistemas ubíquos.

## 1.2 Motivação

Muitas são as diferenças entre um sistema ubíquo e um sistema tradicional (*desktop*, *web*, móvel). Nos sistemas tradicionais, os usuários utilizam dispositivos de entrada como teclado, *mouse* ou *touch screen*. Esses dispositivos são confiáveis, pois não são ambíguos. Além disso, os sistemas tradicionais são orientados a tarefas e desenvolvidos para um único usuário. A interação nesses sistemas corresponde a um diálogo explícito entre o usuário e o computador e geralmente é o usuário que pode começar a interação (POPPE; RIENKS; DIJK, 2007).

No cenário da computação ubíqua, a natureza das aplicações mudou. A seguir são apresentadas quatro grandes tendências em que se pode ver essa mudança, segundo (POPPE; RIENKS; DIJK, 2007).

- Novas possibilidades de sensoriamento (Novas tecnologias de sensoriamento permitem o projeto de sistemas que vão além do *mouse* e teclado tradicional. Reconhecimento de voz, reconhecimento de expressões faciais, reconhecimento da localização são exemplos que tornam as interfaces mais naturais);
- Mudança na iniciativa da interação (Nos sistemas tradicionais, o diálogo é explícito e é o usuário quem o começa. Nos sistemas ubíquos, o diálogo pode ser iniciado pelo próprio sistema, dado à sua capacidade de perceber o usuário, seu ambiente e suas necessidades);
- Diversificação de interfaces físicas (As formas físicas da interface estão se diversificando. Uma direção é construir interfaces maiores, como *displays* interativos. Outra direção é construir interfaces embarcadas, essa é destacada pela popularização de aplicações e dispositivos móveis); e
- Mudanças nos propósitos (Os sistemas tradicionais são em geral baseados em tarefas de trabalho, e seu objetivo chave é a produtividade. Sistemas ubíquos são mais focados na vida diária das pessoas, seu intuito é auxiliar as atividades cotidianas do usuário).

Essas diferenças consideráveis entre o campo da computação tradicional e a computação ubíqua têm apresentado dificuldades para os pesquisadores que tentam utilizar diretamente suas metodologias de avaliação (SCHOLTZ; CONSOLVO, 2004). Existe uma necessidade de novas metodologias, bem como medidas de *software*, que reflitam as características específicas dos sistemas ubíquos.

## 1.3 Objetivos

Considerando o que foi apresentado nas seções 1.1 e 1.2, este trabalho de dissertação tem como objetivo geral propor medidas de *software* para avaliação de características de IHC específicas e essenciais para os sistemas ubíquos<sup>1</sup>.

Para atingir o objetivo geral, os seguintes objetivos específicos são definidos:

1. Investigar características de qualidade e medidas de software para IHC em sistemas ubíquos existentes na literatura;
2. Descrever uma lista final de características relacionadas à IHC em sistemas ubíquos;
3. Selecionar características específicas para IHC em sistemas ubíquos que serão alvo da definição de medidas;
4. Definir as medidas de *software* para avaliar as características selecionadas; e
5. Validar as medidas definidas através de seu uso em sistemas ubíquos e analisar os resultados obtidos.

## 1.4 Metodologia

O desenvolvimento desse trabalho é apoiado pela utilização de metodologias científicas para alcançar os objetivos específicos citados na Seção 1.3. A Figura 1.1 apresenta um mapeamento entre o que é necessário para alcançar o objetivo específico e as metodologias e atividades que foram utilizadas para alcançá-lo.

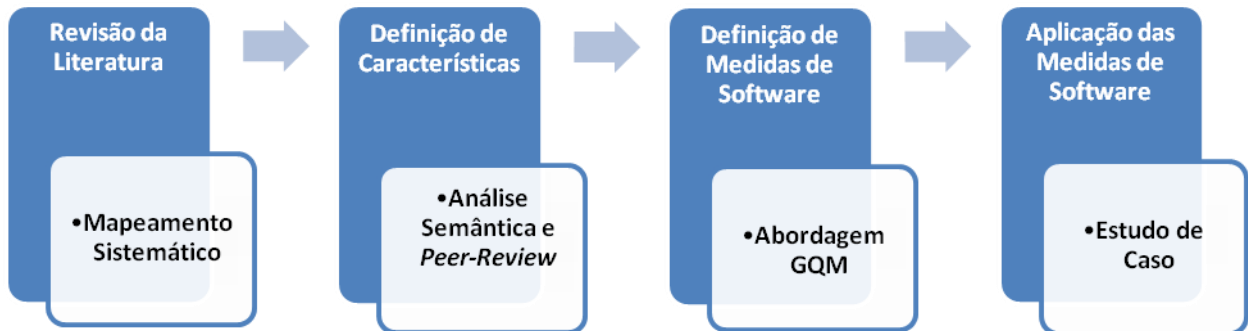
Para o primeiro objetivo específico é necessário realizar uma revisão da literatura, para tanto é utilizada a metodologia chamada Mapeamento Sistemático (MS). Essa metodologia permite realizar uma extensa e sistemática revisão bibliográfica para identificar características de qualidade e medidas de *software* existentes. Todo esse processo é documentado, possibilitando aos leitores o acesso ao seu rigor e sua repetitividade.

Para o segundo objetivo específico, uma Análise Semântica das características de qualidade encontradas pelo MS é realizada para agrupar características com o mesmo significado mas possuem nomes diferentes e retirar características duplicadas. Após isso, uma Revisão em Pares (*Peer Review*) é realizada para avaliar o resultado dessa análise semântica a fim de descrever uma

---

<sup>1</sup> Esse objetivo está inserido em um projeto de pesquisa chamado Maximum - (Uma Abordagem baseada em Medições para Avaliação da Qualidade da Interação Humano-Computador em Sistemas Ubíquos), financiado pela Fundação Cearense de Apoio Tecnológico e Pesquisas (FUNCAP) com o número de processo INC-0064-00012.01.00/12 e de parceria com a Universidade de Valenciennes et du Hainaut-Cambrèsis, localizada na cidade de Valenciennes na França

lista final de características pertinentes para avaliação da IHC em sistemas ubíquos. Após isso, uma seleção de características essenciais e específicas de IHC para os sistemas ubíquos (terceiro objetivo específico) é realizada.



**Figura 1.1** – Metodologia de Trabalho

No quarto objetivo específico, o método *Goal-Question-Metric* (GQM) é utilizado para definir um conjunto de medidas de *software* apropriadas para a avaliação das características definidas e selecionadas no objetivo específico anterior. Esse método foi escolhido devido a experiência que o grupo de pesquisa possui no seu uso para definição de medidas de *software* (RAMOS; OLIVEIRA; ANQUETIL, 2004a, 2004b).

Para o quinto e último objetivo, o Estudo de Caso (EC) é escolhido para aplicar<sup>2</sup> as medidas de *software* definidas no objetivo anterior. EC é uma metodologia de pesquisa empírica que investiga um determinado fenômeno em um contexto.

## 1.5 Organização da Dissertação

Além deste capítulo introdutório, esta dissertação está estruturada em mais cinco capítulos.

- **Capítulo 2 – Fundamentação Teórica:** tem por objetivo fornecer o embasamento teórico acerca das áreas que compõem esse trabalho: Sistemas Ubíquos, Interação Humano-Computador e Qualidade de Software.
- **Capítulo 3 – Investigação e Definição de Características de Qualidade para IHC em Sistemas Ubíquos:** apresenta a metodologia utilizada para encontrar e descrever uma lista de características de interação humano-computador em sistemas ubíquos.

<sup>2</sup> Nessa dissertação, o termo “aplicação” das medidas corresponde a sua avaliação ou validação.

- **Capítulo 4 – Medidas de *Software* para Avaliação da IHC em Sistemas Ubíquos:** apresenta as medidas de *software* propostas para avaliação da qualidade da IHC em sistemas ubíquos. Também apresenta a metodologia utilizada para a seleção e definição das medidas.
- **Capítulo 5 – Estudo de Caso:** apresenta os resultados obtidos no estudo de caso realizado para aplicar as medidas propostas em três aplicações ubíquas.
- **Capítulo 6 – Conclusão:** apresenta os resultados alcançados neste trabalho, seus futuros direcionamentos e a produção bibliográfica alcançada durante o mestrado.

# 2 Fundamentação Teórica

Neste capítulo é apresentada uma revisão bibliográfica das áreas relacionadas a esta pesquisa, a qual serve como base para o entendimento dos desafios e soluções apontados neste trabalho. A Seção 2.1 apresenta definições, características e desafios da área de Sistemas Ubíquos. A Seção 2.2 apresenta os conceitos da disciplina Interação Humano-Computador e, finalmente, a Seção 2.3 apresenta os principais conceitos da área de Qualidade de Software que são utilizados neste trabalho.

## 2.1 Sistemas Ubíquos

### 2.1.1 Definições e Exemplos

A crescente melhoria das tecnologias de hardware (*e.g.*, *smartphones*, *tablets*, sensores, eletrodomésticos com processamento computacional e poder de comunicação) tem possibilitado relevantes avanços no desenvolvimento de sistemas ubíquos (LIMA et al., 2011). Além da visão de Mark Weiser sobre o que são sistemas ubíquos mencionada na Seção 1.1, outras definições são encontradas na literatura a fim de se obter um conceito mais preciso e completo:

- “Computação ubíqua representa o conceito de computação em todo o lugar, fazendo com que o uso da computação e a comunicação sejam transparentes para o usuário” de (YAU; WANG; KARIM, 2002).
- “Computação ubíqua é o uso de um conjunto de computadores dos mais variados tamanhos, formatos e funções, que de forma coordenada e autônoma, auxiliam as pessoas na realização das diversas tarefas cotidianas. Esse auxílio é realizado de tal forma que a infraestrutura computacional responsável fica escondida no ambiente.” de (LIMA, 2011).
- Spínola *et. al.* (2007) realizaram uma revisão sistemática com o objetivo de conceituar a computação ubíqua e suas características. Seu trabalho chegou à seguinte definição: “A computação ubíqua está presente quando serviços computacionais tornam-se disponíveis para os usuários de tal forma que o computador não é mais uma ferramenta visível ou essencial para acessar esses serviços. Assim, os serviços

computacionais são acessados em qualquer tempo ou lugar, de forma transparente, através do uso de dispositivos comuns” (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007).

Além disso, é importante citar que alguns autores consideram a computação ubíqua igual a computação pervasiva ou mesmo a computação móvel, embora sejam conceitos diferentes (ARAUJO, 2003). Por um lado, a computação móvel representa a capacidade do usuário se deslocar com um dispositivo computacional executando um determinado serviço e ainda mantendo conexão, proporcionando a locomoção dos serviços entre ambientes. Por outro lado, a computação pervasiva representa a capacidade dos dispositivos serem embutidos no ambiente físico para obtenção de informações. Logo, a computação ubíqua representa a junção da mobilidade da computação móvel com a integração no ambiente físico que a computação pervasiva proporciona (ARAUJO, 2003).

De acordo com (ARAUJO, 2003) são exemplos de sistemas ubíquos:

- Residências Inteligentes: Sistemas ubíquos para residências objetivam conhecer as atividades dos moradores e fornecer serviços que aumentem sua qualidade de vida, por exemplo, automatizar o controle de luz, tocar música baseado na localização e preferência do usuário e eletrodomésticos inteligentes, como geladeira que detecta a ausência de determinados alimentos importantes para os usuários;
- Guias de Visitas: Objetivam guiar usuários durante uma visita através de dispositivos móveis que possuem serviços como exibir automaticamente o local onde o usuário está e apresentar informações sobre o local, sem que seja necessário o usuário explicitamente interagir com o dispositivo móvel;
- Trabalho Colaborativo: Oferecem suporte a reuniões de trabalho, como registro de anotações feitas em lousas, vídeos da reunião, slides e o áudio de cada participante da reunião sem que interrompa o processo de colaboração; e
- Exemplos de outros domínios, tais como, domínios automotivos, onde carros possuem serviços de alertas sobre o trânsito e sugestão de rotas e domínios de supermercados, onde os carrinhos detectam os alimentos inseridos e fornecem diversas informações em tempo real.

Pelos exemplos percebe-se que nesse paradigma, os sistemas ubíquos estão em todos os lugares e embarcados nos mais variados objetos do dia-a-dia e a interação dos usuários com eles deve ser tão natural quanto utilizar a energia elétrica, ou seja, sem perceber que existe uma



infraestrutura computacional por trás (ROCHA et al., 2011).

## 2.1.2 Características

De acordo com (LIMA, 2011), para alcançar os propósitos citados na seção anterior, é necessário que os sistemas ubíquos levem em consideração as seguintes características:

- Onipresença dos serviços - os usuários podem se mover nos limites físicos do ambiente ou trocar de dispositivos com a sensação de que estão “carregando” os serviços com eles;
- Captura de experiências e intenções - habilidade de perceber o ambiente, as atividades e necessidades do usuário, sem que seja necessário determiná-las explicitamente;
- Adaptação ao Comportamento - habilidade de se adaptar ao contexto e prover serviços e/ou informações relevantes para os usuários;
- Descentralização - as responsabilidades devem ser distribuídas entre vários dispositivos no ambiente, onde cada um é responsável pela execução de um conjunto de funções e tarefas;
- Descoberta de Serviços - os serviços requisitados são descobertos proativamente sem intervenções do usuário;
- Heterogeneidade de Serviços e Dispositivos - habilidade de possuir uma grande variedade de dispositivos e serviços;
- Interoperabilidade - capacidade de diferentes tipos de dispositivos e serviços interajam automaticamente sem a intervenção do usuário;
- Mínima intervenção do usuário - os dispositivos eletrônicos que compõem o ambiente ubíquo devem estar o mais longe possível da percepção dos usuários, com o principal objetivo de não requisitar qualquer operação direta sobre eles; e
- Tolerância à falha - a habilidade que o sistema possui de corrigir a si próprio depois da ocorrência de falhas sem envolver os usuários nesse processo.

Diante dessas características, pode-se perceber que o usuário é citado na maioria delas. Segundo (ROCHA et al., 2011), a computação ubíqua muda o interesse antes centrado na tecnologia, para dar mais atenção ao usuário. O desenvolvimento desses sistemas vai além de aplicações executadas no desktop. Eles procuram atingir o cotidiano dos usuários, no qual o

principal desafio é tornar a computação útil e invisível em diversas situações do dia-a-dia.

Para atender muitas das características citadas anteriormente (captura de experiências e intenções, adaptação, descoberta de serviços automática, heterogeneidade de dispositivos e serviços, mínima intervenção do usuário e tolerância a falhas), o sistema deve ser sensível ao contexto. Segundo (LIMA, 2011) é impossível conceber um sistema ubíquo sem ser sensível ao contexto.

### 2.1.3 Sensibilidade ao Contexto

De acordo com Dey e Abowd (1999), as definições existentes de sensibilidade ao contexto podem ser divididas em duas categorias. A primeira diz respeito às aplicações que utilizam o contexto e a segunda às aplicações que se adaptam ao contexto.

A primeira categoria significa que para um sistema ser sensível ao contexto, ele não precisa, necessariamente, adaptar seu comportamento, basta ser capaz de coletar informações de contexto e armazená-las. Um exemplo de sistema que se enquadra nessa categoria é o aplicativo *PhotoMap* (VIANA et al., 2008), que captura informações contextuais de uma foto, como local em que a foto foi tirada, pessoas presentes na foto, data e horário em que a foto foi retirada, para armazená-las como *tags* na foto. Isso facilita buscá-las em um momento posterior, bem como compartilhá-las com outros usuários.

A segunda categoria significa que o sistema precisa adaptar seu comportamento para ser sensível ao contexto. Neste caso, a aplicação precisa ser capaz de coletar informações de contexto e fornecer serviços de acordo com essas informações coletadas. Por exemplo, uma aplicação para dispositivos móveis pode ser capaz de coletar o ambiente que o usuário se encontra e modificar o perfil das chamadas telefônicas, caso o usuário esteja em um cinema, a aplicação é capaz de modificar o perfil para silencioso.

Visando uma definição mais geral e que permita agregar um maior número de aplicações, Dey e Abowd (1999) definiram sensibilidade ao contexto como: “Um sistema é sensível ao contexto se ele utiliza contexto para prover serviços e/ou informações relevantes para o usuário, onde relevância depende da tarefa do usuário” (DEY; ABOWD, 1999). Contexto é qualquer informação que pode ser usada para caracterizar a situação de uma entidade. Uma entidade pode ser uma pessoa, um lugar, ou um objeto que é considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação (DEY, 2001).

Segundo Dey e Abowd (1999), para ter ciência do contexto, é necessário responder às seguintes perguntas:

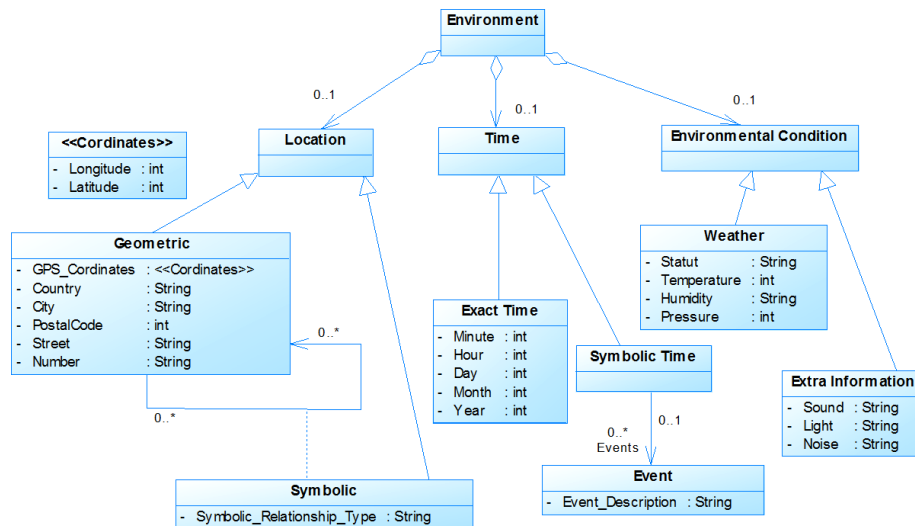
- Quem? - está relacionada com as informações de identidade dos usuários;
- Onde? - está relacionada com a localização do usuário. É uma das questões mais exploradas e o interessante é unir as respostas do “onde” com outras informações contextuais;
- O quê? - está relacionada com a identificação de atividades do usuário;
- Quando? - está relacionada com o contexto temporal; e
- Por quê? - está relacionada com as informações que causaram determinadas ações do usuário. O desafio é entender a razão pela qual o usuário está realizando uma determinada atividade em vez de apenas perceber o que ele/ela está fazendo.

Para exemplificar um serviço ubíquo que identifica alguns desses pontos, pode-se citar um telefone celular de um usuário que está no trabalho (onde) durante uma reunião de negócios (o que) e recebe uma ligação, conseguindo identificar se a pessoa que está ligando é prioridade (quem), por exemplo, é um amigo ou parente muito próximo que está doente (por quê) e decide se o avisa com sinais sonoros ou não.

Essas informações de contexto são obtidas a partir de uma variedade de fontes que são inclinadas a falhas e se diferem na qualidade do contexto obtido, tornando o desenvolvimento de software sensível ao contexto uma tarefa complexa. Para reduzir essa complexidade, o desenvolvimento deve ser apoiado por técnicas apropriadas de raciocínio e modelagem de contexto (BETTINI et al., 2010).

As técnicas existentes de modelagem de contexto se diferem com relação a: facilidade com que as informações do mundo real podem ser obtidas pelos engenheiros de software, o poder de expressividade, o suporte que eles podem prover com relação ao raciocínio das informações de contexto, o desempenho computacional do raciocínio e a escalabilidade do gerenciamento das informações de contexto (BETTINI et al., 2010).

Exemplos dessas técnicas de modelagem de contexto são: modelos chave-valor, modelos baseados em esquemas de marcação, modelos gráficos, modelos orientados a objetos, modelos baseado em lógica, modelos baseado em ontologias (FERNANDES, 2009). Um exemplo de modelo de contexto pode ser visto na Figura 2.1.



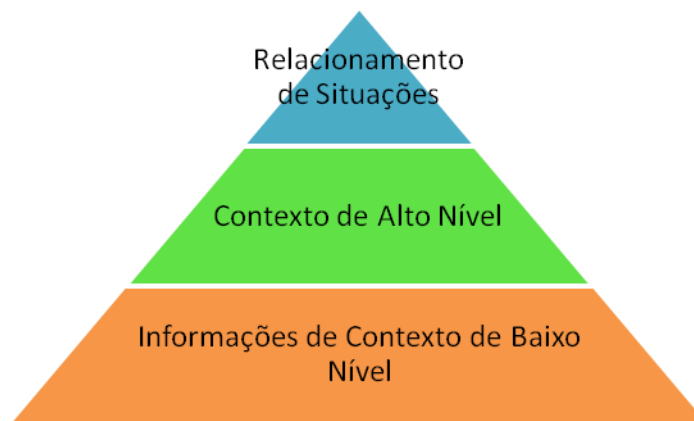
**Figura 2.1** - Modelo de contexto. Fonte: (BACHA; OLIVEIRA; ABED, 2011)

Esse modelo de contexto foi desenvolvido por (BACHA; OLIVEIRA; ABED, 2011) e está modelado como diagrama de classes da UML. Ele descreve informações de contexto sobre o ambiente em que a interação ocorre entre o usuário e o sistema: localização do ambiente (país, cidade, rua, código postal, número), condições ambientais (clima e informações sobre luminosidade, barulho) e tempo (ano, mês, dia, hora, minuto).

Para tornar as aplicações sensíveis ao contexto mais confiáveis e úteis, também é necessário criar abstrações de alto nível a partir dessas informações de contexto. (BETTINI et al., 2010) chamam essas abstrações de Situações de Contexto. Por exemplo, *em\_reunião\_agora* é uma situação de contexto que pode ter sido causada pelos seguintes pedaços de contexto: localização da pessoa e informação da agenda.

A Figura 2.2 apresenta uma visão geral proposta por (BETTINI et al., 2010) para abstrações e interpretação de contexto: as informações de contexto de baixo nível são semanticamente interpretadas por uma camada de contexto de alto nível e são chamadas de informações de contexto crua. Assim, situações, que são as informações de contexto de alto nível, são reutilizadas em diferentes aplicações e ambientes. Os relacionamentos entre situações de contexto são capazes de reduzir o espaço de busca por situações e conhecer possíveis relações (por exemplo, saber as possíveis situações que sucederam a situação atual).

No trabalho de (YE; DOBSON; MCKEEVER, 2012) é possível encontrar uma revisão da literatura sobre várias técnicas de detecção de situações de contexto: programação lógica, lógica temporal e espacial, ontologias, lógica fuzzy, redes bayesianas, modelos de Markov, árvores de decisão, entre outras.



**Figura 2.2** – Camadas de Abstração e Interpretação de Contexto, traduzida de (BETTINI et al., 2010)

Além da preocupação com técnicas de raciocínio e modelagem do contexto, o desenvolvimento do sistema ubíquo necessita também de infraestruturas de suporte ao seu desenvolvimento que levem em consideração as características citadas anteriormente. Algumas arquiteturas dessas infraestruturas são apresentadas na Seção 2.1.4 a seguir.

## 2.1.4 Arquiteturas

Várias infraestruturas de *software* já foram criadas para apoiar o desenvolvimento de sistemas ubíquos (ROMÁN et al., 2002)(DONOHOO; OHLSEN; PASRICHA, 2011)(GRIMM et al., 2004). Essa seção apresenta duas arquiteturas desenvolvidas que facilitam o projeto e implementação de sistemas ubíquos, bem como uma arquitetura que está voltada para aplicações sensíveis ao contexto.

O projeto Aura é uma infraestrutura de *software* que tem como principal objetivo minimizar as distrações na atenção do usuário por meio de um ambiente que se adapta às suas necessidades e contexto. Aura é voltado para ambientes da computação ubíqua que envolvem comunicação sem fio, dispositivos móveis ou vestíveis e espaços inteligentes (GARLAN; SIEWIOREK; STEENKISTE, 2002).

Esse projeto considera como recurso principal, a atenção do usuário, que se refere á habilidade de atender suas tarefas primárias (*e.g.*, andar, dirigir ou outras interações do mundo real), ignorando as distrações causadas pela tecnologia (*e.g.*, desempenho ruim, falhas).

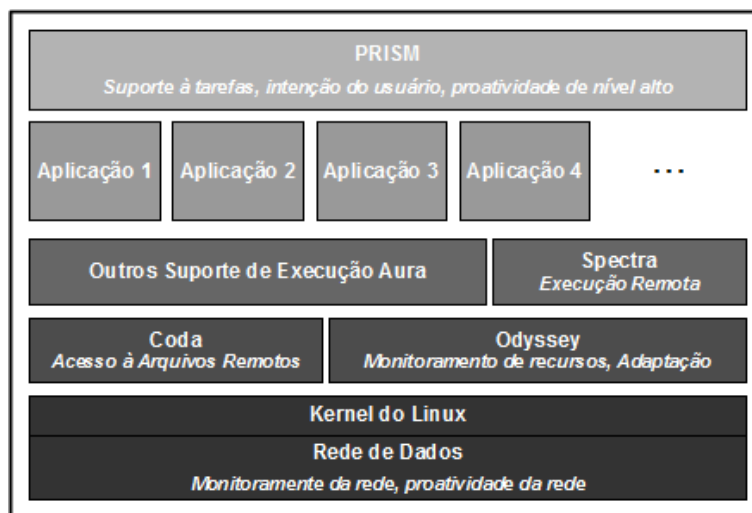
Para alcançar seu objetivo, o Aura abrange todos os níveis de um sistema: do *hardware*, passando pelo sistema operacional até as aplicações e usuários finais. O projeto considera dois conceitos para seu desenvolvimento: proatividade (habilidade de uma camada do sistema antecipar as requisições de uma camada superior) e auto-ajuste (as camadas se adaptam por

observar as demandas feitas sobre elas e ajustam o uso de seus recursos de acordo com isso).

Um exemplo de cenário que o projeto Aura deseja tornar realidade com suas pesquisas é: “Jane está no portão de embarque 23 no aeroporto de Pittsburgh, esperando por seu voo. Ela finalizou a edição de vários documentos importantes e gostaria de utilizar a conexão *wireless* para enviá-los por e-mail. Infelizmente, a largura de banda é baixa porque muitos passageiros nos portões 22 e 23 estão navegando na *web*. Aura observa que, na largura de banda atual, Jane não será capaz de terminar o envio de seus documentos antes de sua partida do voo. Consultando o serviço de largura de banda da rede sem fio do aeroporto e o serviço de programação dos voos, Aura descobre que a largura de banda da rede sem fio é excelente no portão 15, e que não há voos partindo e chegando às portas próximas por meia hora. Uma caixa de diálogo aparece na tela de Jane sugerindo que ela vá ao portão 15, e que é apenas três minutos de distância do portão que está. Ele também pede a ela para priorizar seus e-mails, de modo que o mais crítico seja enviado primeiro. Jane aceita o conselho de Aura e vai até o portão 15. Ela assiste os resultados das eleições na TV que existe no portão de embarque, até que Aura informa que está muito perto de terminar o envio de suas mensagens, assim ela começa a caminhar de volta ao portão 23. A última mensagem é transmitida enquanto ela caminha e, finalmente, ela está de volta ao portão 23 a tempo da chamada de seu voo”.

Neste cenário, Aura está utilizando o conceito de proatividade, já mencionado anteriormente, assim, consegue determinar quanto tempo levará para Jane enviar os seus e-mails e a avisa antecipadamente. Além disso, Aura cruza informações de diferentes camadas de sistemas, determinando tanto o congestionamento da rede sem fio (conhecimento de baixo nível), quanto o tempo de embarque do voo (conhecimento no nível do usuário) para permitir a Jane enviar todos os seus documentos antes de partir. A Figura 2.3 apresenta a arquitetura desenvolvida para apoiar o desenvolvimento desse tipo de sistema.

O componente *Odyssey* apoia o monitoramento de recursos e as adaptações sensíveis ao contexto. O componente *Coda* provê suporte para acesso a arquivos de forma remota, desconectável e adaptável à largura de banda. O componente *Spectra* é um mecanismo de execução remota adaptativa que utiliza informações de contexto para decidir como melhor executar uma chamada remota. A camada *Prism* é responsável por capturar e gerenciar as intenções do usuário. Ela desempenha esse papel através da representação das intenções do usuário por meio de tarefas, que são disponibilizadas para o resto do sistema, formando, assim, uma base para adaptar ou antecipar as necessidades do usuário.

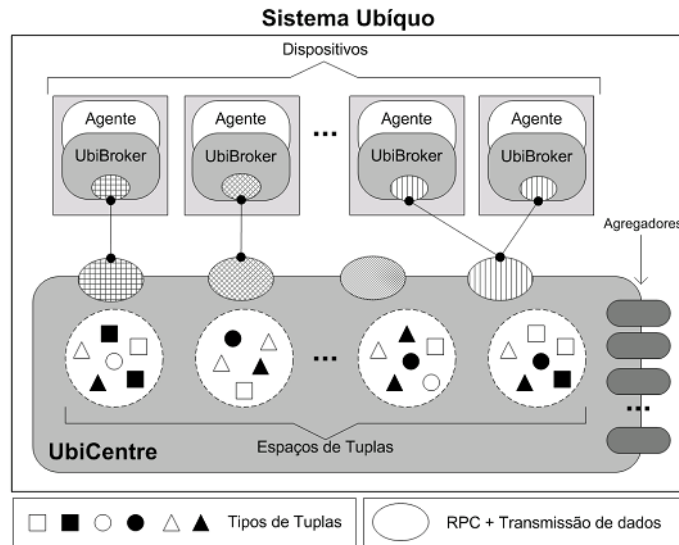


**Figura 2.3** - Arquitetura Aura, traduzida de (GARLAN; SIEWIOREK; STEENKISTE, 2002)

De acordo com (LIMA, 2011), essa arquitetura provê suporte à mínima intervenção do usuário, onipresença dos serviços, autonomicidade e adaptabilidade, no entanto, não menciona soluções para a interoperabilidade, importante requisito no desenvolvimento de sistemas ubíquos. Assim, o trabalho de (LIMA et al., 2011) propõe um sistema de suporte ao desenvolvimento de sistemas ubíquos chamado SysSu, que leva em consideração a interoperabilidade e a coordenação entre agentes no sistema. Esse sistema de suporte é apresentado por meio de uma arquitetura de referência (Figura 2.4).

Nesse trabalho, (LIMA et al., 2011) considera que um sistema ubíquo é formado por vários dispositivos e cada um deles pode possuir vários agentes. Neste caso, agentes representam qualquer entidade de software que está presente no sistema e é capaz de realizar processamento e comunicação. Além de agentes, o dispositivo possui o *UbiBroker*, *middleware* responsável pela comunicação dos agentes com o *UbiCentre*. Esse é o componente que garante a interoperabilidade do sistema, pois cada agente, com diferentes especificações, possuirá um *middleware* capaz de abstrair as diferenças e realizar a comunicação com o *UbiCentre*.

O *UbiCentre* é um repositório de espaços de tuplas centralizado. O espaço de tupla pode servir como repositório de informações contextuais. Esse repositório pode ser acessado concorrentemente através dos *UbiBrokers*. A vantagem de sua utilização está no fato que eles permitem desacoplamento entre os agentes, possibilitando, por exemplo, um agente aparecer e inserir uma tupla (nova informação contextual) e outro agente aparecer e retirar essa tupla sem que seja necessário conhecer o agente que a colocou e nem quando ele o fez. Além de possuir o repositório de espaços de tuplas, o *UbiCentre* fornece um conjunto de operações que podem ser executadas sobre as tuplas.



**Figura 2.4 -** Arquitetura SysSu. Fonte: (LIMA et al., 2011)

Apesar de essas arquiteturas se preocuparem com as características citadas anteriormente, uma arquitetura mais detalhada para representar a sensibilidade ao contexto é necessária. (MARINHO et al., 2010) propõe uma linha de produtos para o desenvolvimento de aplicações móveis e sensíveis ao contexto e apresenta sua arquitetura na Figura 2.5.

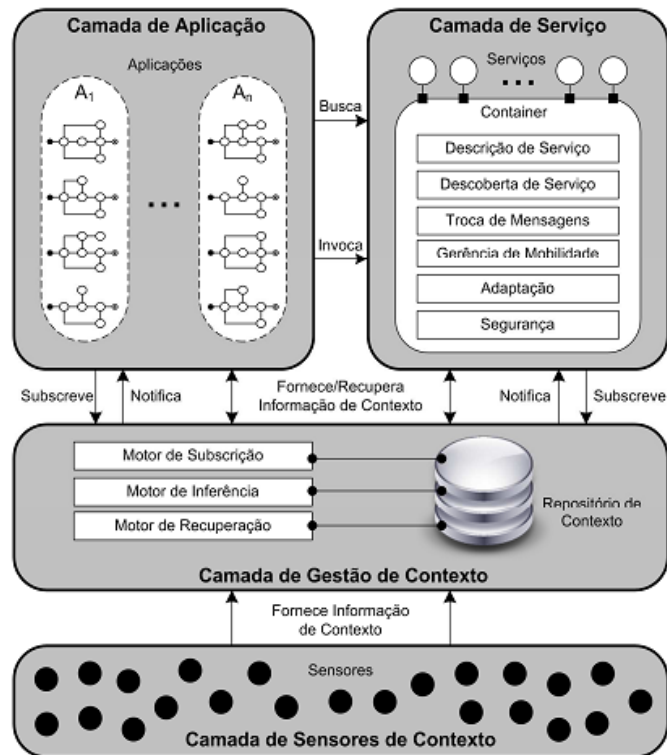
A Camada de Sensores de Contexto é representada por sensores, os quais são capazes de capturar informações contextuais. Tais informações podem ser capturadas de diferentes fontes, *e.g.*, através de um sistema de informações ou diretamente de um dispositivo de *hardware*.

A Camada de Gestão de Contexto é responsável por gerenciar as informações de contexto obtidas pelas camadas de sensores, aplicação e serviço. O Motor de Recuperação é utilizado pelas camadas de aplicação e serviço para recuperar informações que estão no Repositório de Contexto. O Motor de Inferência é responsável por inferir novas informações contextuais a partir de informações existentes no repositório. O motor de subscrição é responsável por notificar quais contextos estão ativos em um determinado instante do tempo para as aplicações e/ou serviços que desejam obter tais contextos.

A Camada de Serviço é responsável pela implantação, gerenciamento e comunicação dos serviços com eles mesmos e com as aplicações. Nessa camada o *Container* é responsável por gerenciar os serviços e permitir a comunicação entre eles e as aplicações. O módulo *Descrição de Serviços* é responsável por registrar as descrições dos serviços, pois cada serviço precisa de uma descrição para que possa ser encontrado. O módulo *Descoberta de Serviços* é responsável por encontrar um serviços de acordo com a descrição requisitada. O módulo *Troca de Mensagens* é responsável por permitir a mobilidade do dispositivo. O módulo de *Gerência de Mobilidade* é



responsável pela mobilidade do código e, por fim, o módulo *Segurança* é responsável pela mobilidade do usuário.



**Figura 2.5** - Arquitetura do Mobiline. Fonte: (MARINHO et al., 2010)

A Camada de Aplicação é composta pelas aplicações móveis e sensíveis ao contexto e é responsável pela descrição e execução das aplicações. Uma aplicação pode ser composta de vários serviços da camada de serviço.

As arquiteturas apresentadas acima são abordagens para melhorar a modularização do software. Isso, de acordo com (ROCHA et al., 2011), representa um dos desafios de pesquisa na área de sistemas ubíquos. Tais desafios são apresentados na Seção 2.1.5 a seguir.

## 2.1.5 Desafios de Pesquisa

O trabalho de (ROCHA et al., 2011) apresenta cinco grandes desafios sobre o desenvolvimento de sistemas ubíquos:

- **Modularização de Software:** a modularização é um atributo importante no desenvolvimento de sistemas. Tal atributo permite que o sistema seja decomposto em vários módulos a fim de diminuir sua complexidade e isso é extremamente essencial no desenvolvimento de sistemas complexos como sistemas ubíquos. A questão é que o módulo nos sistemas ubíquos pode evoluir para abstrações maiores. Por exemplo, o

módulo pode ser representado por um domínio de software. Logo, o seguinte questionamento surge: quais modelos de abstração de módulo seriam os mais adequados para tratar a modularização em sistemas ubíquos?

- **Desenvolvimento Globalizado:** o desenvolvimento de software tem se tornado globalizado, o que traz diversos problemas. Para solucioná-los, diversas pesquisas têm sido realizadas em gerência de projetos, gerência de configuração, ferramentas de apoio, entre outras. O desenvolvimento de sistemas ubíquos também segue essas tendências, porém a complexidade é muito maior, dado que sistemas ubíquos devem ser desenvolvidos em diferentes ambientes, plataformas e *hardwares*. Uma empresa que deseja desenvolver um sistema ubíquo deverá lidar com diferentes empresas fabricantes de *software* e *hardware*, o que exigirá uma adequação de soluções existentes, como gerências de configuração, ferramentas e processos de desenvolvimento.
- **Dependabilidade:** a dependabilidade é uma propriedade que pode ser decomposta em confiabilidade, segurança, confidencialidade, integridade e manutenibilidade. Os sistemas ubíquos impõem novos desafios à dependabilidade, devido às suas características (heterogeneidade, descentralização e mobilidade). Eles apresentam falhas, difíceis de serem previstas em tempo de projeto e, além disso, é difícil implementar mecanismos de prevenção e tratamento de falhas em sistemas altamente distribuídos. Esses problemas exigem que novas técnicas de engenharia de software sejam desenvolvidas para essa área.
- **Interoperabilidade:** o sistema ubíquo é composto de vários sistemas. Cada subsistema que o compõe pode ser de diferentes linguagens de programação e sistemas operacionais. Além disso, eles são dinâmicos, podendo entrar e sair do ambiente a qualquer momento. Assim, um subsistema deve estar apto a conseguir se comunicar com outro diferente subsistema. A solução tradicional para tratar as diferenças deve ir além de sistemas baseados em *middlewares*, para novas soluções que sejam em tempo de execução e que permitam a comunicação entre diferentes subsistemas.
- **Segurança:** os sistemas ubíquos são abertos à integração de serviços e dispositivos de terceiros. Devido a isso, não é possível conhecer em tempo de projeto todos os elementos que farão parte do sistema, assim, o sistema fica sujeito a usos indevidos, colocando em risco sua segurança e privacidade. Assim, é necessária infraestruturas de softwares que se preocupem com aspectos de segurança.

## 2.2 Interação Humano-Computador

### 2.2.1 Definições

A IHC é uma disciplina interessada com o projeto, implementação e avaliação de sistemas computacionais interativos para uso humano e no estudo dos principais fenômenos que o cercam (CHAIRMAN-HEWETT, 1992). De acordo com Barbosa e Silva (2010), a interação e a interface devem ser adequadas para que os usuários possam utilizar o sistema da melhor maneira possível. Essa autora define os seguintes critérios de qualidade de uso que a interação deve ter para ser considerada adequada: Usabilidade, Experiência do Usuário, Acessibilidade e Comunicabilidade.

Usabilidade é o mais conhecido e utilizado deles e está relacionado com a facilidade de aprendizado e uso do sistema. Nielsen (1994) define o critério de qualidade Usabilidade como um conjunto de fatores que qualificam quão bem um usuário pode interagir com um sistema computacional. Tais fatores são: facilidade de aprendizado, facilidade de memorização, eficiência, segurança no uso e satisfação do usuário (NIELSEN, 1994).

- Facilidade de aprendizado diz respeito ao tempo e esforços necessários para o usuário aprender a utilizar o sistema. Por exemplo, pode-se avaliar quanto tempo um usuário leva para aprender a realizar um conjunto mais amplo de atividades.
- Facilidade de memorização diz respeito ao esforço necessário para lembrar como interagir com a interface do sistema interativo, conforme aprendido anteriormente. Por exemplo, pode-se avaliar qual o desempenho do usuário, após um período de tempo sem utilizar o sistema.
- Eficiência diz respeito ao tempo necessário para conclusão das atividades com o apoio tecnológico. Tal componente leva em consideração que a produtividade deve ser alta quando o usuário já aprendeu a utilizar o sistema.
- Segurança no uso diz respeito ao grau de proteção de um sistema contra condições desfavoráveis ou até mesmo perigosas para os usuários. Uma forma de evitar erros, por exemplo, pode ser não colocar botões de “remover tudo” próximo a um botão de “gravar”.
- Satisfação do usuário diz respeito ao sistema ser agradável para uso de modo que o usuário se sinta satisfeito ao utilizá-lo.

Existem outras definições de usabilidade. A definição, segundo a norma ISO 9241 é o

grau em que um produto pode ser usado por usuários para alcançar objetivos com eficácia, eficiência e satisfação em um contexto de uso específico. Sendo que eficácia diz respeito à capacidade dos usuários atingirem corretamente seus objetivos, eficiência diz respeito aos recursos necessários para atingi-los e satisfação está relacionada com uma boa experiência de uso (ISO 9241-11, 1998).

O critério de qualidade Experiência do Usuário representa uma atenção maior para as emoções e sentimentos dos usuários, no entanto, alguns autores apenas a consideram como uma atenção maior à satisfação do usuário, componente da usabilidade (BARBOSA; SILVA, 2010).

O terceiro critério de qualidade, segundo (BARBOSA; SILVA, 2010), é a Acessibilidade, que é a capacidade de remover barreiras que impedem uma quantidade maior de pessoas com diferentes capacidades de utilizar o *software*. Garantir a acessibilidade permite que mais usuários utilizem um determinado sistema. Por exemplo, um leitor de tela pode apoiar um usuário com deficiência visual na interação com um sistema.

O critério de comunicabilidade diz respeito à responsabilidade do designer do sistema comunicar aos usuários a lógica do comportamento do sistema, acredita-se que se o usuário for capaz de entender a lógica utilizada na concepção de um sistema, ele terá mais chances de utilizá-lo com criatividade e produtividade. Um exemplo de comunicabilidade está no uso de analogias, projetando a interface semelhante ao que já existe no mundo real.

Os critérios de qualidade em uso podem ser avaliados através de diversos métodos de avaliação da IHC. Tais métodos de avaliação são apresentados na Seção 2.2.2 a seguir.

## 2.2.2 Métodos de Avaliação

(BARBOSA; SILVA, 2010) classifica os métodos nas seguintes categorias: métodos de investigação, de inspeção e observação de uso. Tais métodos são descritos abaixo.

### Métodos de Investigação

Esses métodos permitem aos avaliadores coletarem opiniões e expectativas do usuário com relação a um sistema. Eles são utilizados para entender uma determinada situação, identificar problemas e soluções. Alguns exemplos são entrevistas e questionários.

Entrevistas são conversas guiadas por um roteiro que contém perguntas e/ou tópicos para guiar um entrevistador ao coletar dados de um entrevistado. Elas podem ser classificadas em estruturadas, não estruturadas e semiestruturadas. Nas estruturadas, o entrevistador realiza perguntas previamente definidas, em geral são compostas por perguntas com respostas fechadas. Nas entrevistas não estruturadas, a maioria das perguntas são abertas, por outro lado, as entrevistas semiestruturadas são um meio termo, pois contêm tanto perguntas abertas, quanto fechadas.

Questionário é um método de avaliação muito utilizado, trata-se de formulários com perguntas que os usuários devem responder. Existem diversos tipos de perguntas e respostas, dentre os quais são destacadas: múltipla escolha, faixas de valores, perguntas abertas e escalas.

Questões de múltipla escolha permitem que o usuário escolha uma ou mais opções de resposta já definidas. Questões de faixas de valores permitem o usuário escolher entre várias faixas de valores (*e.g.*, faixa etária: 21-30 anos, 31-40 anos).

Perguntas abertas são usadas para obter informações livres e mais detalhadas sobre um determinado ponto. As questões com escalas são utilizadas para facilitar a comparação de respostas com usuários. As mais conhecidas escalas são a de Likert e as escalas de diferenciais semânticos (exemplos destes podem ser vistos na figura abaixo).

<p><b>É fácil realizar o cadastro no site.</b></p> <ul style="list-style-type: none"><li><input type="radio"/> concordo plenamente</li><li><input type="radio"/> concordo parcialmente</li><li><input type="radio"/> não concordo nem discordo</li><li><input type="radio"/> discordo parcialmente</li><li><input type="radio"/> discordo totalmente</li></ul>	<p><b>Para cada par de adjetivos a seguir, marque o valor correspondente à sua opinião sobre a página de um produto do site:</b></p> <p>atraente <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> feia</p> <p>clara <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> confusa</p> <p>útil <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> <input type="radio"/> inútil</p>
--	---

**Figura 2.6** - Exemplo da Escala de Likert e Diferencial Semântico. Fonte: (BARBOSA; SILVA, 2010)

## Métodos de Inspeção

Esses permitem ao avaliador investigar uma solução para tentar encontrar problemas de interação do usuário com o sistema. Eles não envolvem o usuário e sim um especialista em avaliações e podem ser utilizados ao longo de todo processo de design, somente é necessário que exista uma representação do sistema. Alguns métodos são avaliação heurística e percurso cognitivo, descritos a seguir.

Avaliação heurística é um método de avaliação criado para encontrar problemas de usabilidade. Tal método é executado por avaliadores com auxílio de um conjunto de diretrizes de usabilidade, que especificam características desejáveis da interação e da interface, chamadas por Nielsen (1994) de heurísticas. (NIELSEN, 1994) descreve um conjunto inicial de heurísticas para avaliação de um sistema, exemplos de suas heurísticas são:

- Visibilidade do estado do sistema, onde o usuário deve sempre ser informado sobre o status do sistema;
- Correspondência entre o sistema e o mundo real, onde o sistema deve ter um vocabulário familiar para o usuário;
- Controle e liberdade do usuário, onde o sistema deve permitir ao usuário fazer e desfazer ações;
- Consistência e padronização, onde o sistema deve seguir um padrão com o intuito de não confundir o usuário;
- Flexibilidade e eficiência no uso, onde o sistema deve possuir aceleradores, como botões de comando e teclas de atalho, para tornar a interação mais eficiente.

As heurísticas podem ser expandidas para incluir novas diretrizes que reflitam determinados tipos de software (*e.g., desktop, web*) e/ou domínios de software (*e.g., educação, trabalho colaborativo*).

As atividades necessárias para realizar a avaliação heurística são: **preparação**, na qual os avaliadores estudam o software e selecionam as partes dele que serão avaliadas, bem como as heurísticas que serão utilizadas. A **coleta e interpretação de dados**, na qual os avaliadores inspecionam a interface para encontrar problemas, e, caso os encontrem, armazenam-os, indicando o local do sistema que foi encontrado, a gravidade e uma recomendação de solução. A **consolidação dos resultados**, a qual os avaliadores se reúnem e revisam os problemas encontrados e, por fim, **o relato dos resultados**, no qual os avaliadores geram um relatório com

os resultados encontrados e consolidados por todos os avaliadores.

Percurso cognitivo possui como principal objetivo avaliar a facilidade de aprendizado, através da exploração da interface pelas tarefas do usuário. Nesse método, o avaliador percorre a interface executando as ações necessárias para realização de uma tarefa, a cada ação o avaliador se coloca no lugar do usuário e detalha como seria a interação. As atividades necessárias para executar esse método de avaliação são:

- **Preparação**, na qual é identificado o perfil do usuário, as tarefas que farão parte da avaliação e as ações para executar cada tarefa. As tarefas a serem avaliadas podem ser representadas por modelos de tarefas. Um exemplo muito conhecido para descrever tarefas é o GOMS (*Goal, Operators, Methods, and Select Rules*). Esse método especifica objetivos (*goals*), operadores (*operators*), métodos (*methods*) e regras de seleção (*selection rules*). *Objetivos* representa o que o usuário deseja realizar com o software. *Operadores* representam ações concretas que o software permite executar. *Métodos* são sequências de subobjetivos e operadores necessários para realizar um objetivo e regras de seleção são utilizadas quando existe mais de um método. Além de representar detalhadamente as tarefas, o método GOMS também pode ser utilizado para prever o desempenho de uma tarefa.
- **Coleta de Dados**, o avaliador percorre o sistema executando as ações necessárias para realizar as tarefas. Para cada ação o avaliador se pergunta às seguintes questões: o usuário vai tentar atingir o efeito correto? O usuário vai notar que a ação correta está disponível? O usuário vai associar a ação correta com o efeito que está tentando atingir? Se a ação for executada corretamente, o usuário vai perceber que está progredindo na direção de concluir a tarefa?
- **Interpretação**, o avaliador deve sumarizar uma história sobre o sucesso ou falha na interação.
- **Resultados**, o avaliador deve apresentar os resultados em forma de relatório com os problemas encontrados e sugestões de solução.

### **Métodos de Observação em Uso**

Tais métodos consistem em coletar dados sobre a utilização de sistemas por usuários. Eles podem ser feitos em laboratório ou em campo. A observação em campo permite coletar dados sobre o uso de sistemas no mundo real, obtendo, assim, resultados mais ricos sobre a experiência do usuário. Já a observação em laboratório consiste em coletar dados de uso em um

ambiente controlado. O mais conhecido e utilizado deles é o teste de usabilidade, que está descrito a seguir.

O Teste de Usabilidade visa avaliar a usabilidade de um sistema através do seu uso por um grupo de usuários alvo do sistema. Os objetivos do teste de usabilidade são definidos através de quais componentes de usabilidade se deseja avaliar. Por exemplo, o objetivo pode ser avaliar a eficiência da interação. Dado que os objetivos estão definidos, cada um deles é acompanhado de perguntas associadas a dados mensuráveis: quanto tempo o usuário demora em executar uma determinada tarefa? As atividades para realização do teste de usabilidade são:

- **Preparação**, na qual o avaliador define o perfil de usuário, define os objetivos de avaliação, prepara o ambiente e o material para coletar e armazenar os dados necessários e define uma lista de tarefas que devem ser executadas no teste.
- **Coleta de dados** inclui o questionário pré-teste, a execução das tarefas pelos usuários e a entrevista pós-teste. Dependendo de qual componente foi escolhido para se medir, os dados, como “grau de sucesso da execução de uma tarefa”, “tempo para realização” e “grau de satisfação”, são coletados durante o uso do sistema. Além disso, observações feitas pelo avaliador, vídeos da interação e áudio são coletados.
- **Interpretação e consolidação dos resultados** implica em organizar os dados de forma que eles possam responder se os objetivos definidos anteriormente foram atingidos. Importante mencionar que além de dados quantitativos, o teste de usabilidade também pode fornecer dados qualitativos. A análise conjunta desses dois tipos de dados pode fornecer resultados que não seriam observados utilizando somente um tipo de dado.
- **Relato dos resultados**, no qual um relato de como foi realizado o teste de usabilidade e seus resultados deve ser feito.

### 2.2.3 IHC em Sistemas Ubíquos

No campo da computação ubíqua, a IHC objetiva que os sistemas devem ser projetados para apoiar a interação implícita (POSLAD, 2009). A definição para interação implícita é “uma ação executada pelo usuário que não é essencialmente destinada a interagir com um sistema computadorizado, mas que tal ação o sistema compreende como entrada” (SCHMIDT, 2000).

Schimdt (2000) apresenta um exemplo de interação implícita. Quando um usuário entra em um ambiente escuro, ele precisa ligar manualmente a luz. Porém, um sistema inteligente



percebe que o usuário entrou no ambiente e liga a luz automaticamente. Esse tipo de interação é definido como interação implícita, ou seja, é o oposto da interação explícita, no qual o usuário precisa especificar cada detalhe da interação (SCHMIDT, 2000). Isso requer que os sistemas sejam confiáveis e precisos para detectarem que o usuário, seu contexto de uso e sua capacidade de se adaptar estejam de acordo com as suas operações.

Para alcançar os objetivos da interação implícita, o sistema deve ser capaz de entender o comportamento do usuário e se adaptar a ele. Nesse cenário, a palavra chave é sensibilidade ao contexto, já apresentado na Seção 2.1.3 (DEY; ABOWD, 1999).

A interação implícita coloca o usuário no centro dos sistemas, de modo que seu controle responda para o usuário e seja conduzido externamente por ações implícitas dele. Ao contrário da interação humano-computador explícita, ela reduz ou até mesmo remove o controle explícito do usuário (VIEIRA et al., 2009).

Um dos benefícios da interação baseada em contexto é evitar sobrecarregar o usuário com muitas informações e tomada de decisões. Isso corresponde à visão original da Computação Ubíqua, que é fazer o acesso a serviços mais transparente e menos evasivo (POSLAD, 2009).

Além do aspecto de sensibilidade ao contexto, a IHC na computação ubíqua deve considerar interfaces mais naturais no projeto de sistemas (POSLAD, 2009). As interfaces de computador que suportam formas de comunicação mais naturais estão começando a complementar ou substituir os elementos do paradigma de interação com interface gráfica do usuário. Interfaces baseadas em voz, gestos e telas sensíveis ao toque são exemplos de interfaces que tornam a interação mais natural (ABOWD; MYNATT; RODDEN, 2002).

Outro aspecto importante da computação ubíqua é que ela tentará mesclar suavemente artefatos computacionais com o mundo dos artefatos físicos (ABOWD; MYNATT; RODDEN, 2002). Salas, geladeiras, mesas, casas deverão ser projetadas de tal forma que a interação seja a mais natural possível.

## **2.2.4 Dificuldades na Avaliação da IHC em Sistemas Ubíquos**

Segundo Posland (2009), além das dificuldades de desenvolvimento que a computação ubíqua possui por si só, apresentados na Seção 2.1.5, no campo de Interação Humano-Computador (IHC) para sistemas ubíquos a complexidade da avaliação aumenta como consequência dos diversos cenários descritos abaixo:

- Dificuldade em disponibilizar um sistema ubíquo no mundo real e ainda coletar e

armazenar dados de interação contínua de forma que não atrapalhe a interação do usuário, visto que a interação na computação ubíqua deve ser o mais invisível possível;

- Complexidade em lidar com tarefas que podem ser suspensas e retomadas, compartilhadas entre diversos participantes e/ou que os usuários podem executar diversas atividades concorrentemente e em vários ambientes;
- Complexidade em determinar com precisão e confiabilidade o contexto do usuário por causa do não determinismo do indivíduo, o usuário pode não estar certo do que quer fazer e também por causa do não determinismo do seu ambiente.

A computação ubíqua muda completamente a forma como as pessoas interagem com os computadores. Os serviços computacionais deverão estar em toda parte, auxiliando os usuários nas mais diversas atividades do dia-a-dia. Porém, para se tornar um sucesso, as aplicações ubíquas precisam ser projetadas priorizando aspectos da IHC (Riekki and Isomursu, 2004).

## 2.3 Qualidade de Software

### 2.3.1 Definições

Ao longo dos anos pesquisadores e organizações tem definido o termo qualidade de *software*. Para a (ISO/IEC 9126, 2001), “Qualidade é a totalidade de características e critérios de um produto ou serviço que exercem suas habilidades para satisfazer às necessidades declaradas ou envolvidas”. Para (PRESSMAN, 2009) “Qualidade de *software* é a conformidade a requisitos funcionais e de desempenho que foram explicitamente declarados, a padrões de desenvolvimento claramente documentados, e a características implícitas que são esperadas de todo software desenvolvido por profissionais”. A ISO 25000 define qualidade como a capacidade do produto de *software* satisfazer necessidades explícitas e implícitas quando utilizado sobre condições específicas (ISO/IEC 25000, 2005).

A partir dessas definições é possível observar alguns aspectos importantes sobre Qualidade. Um deles é o de satisfação dos usuários, que pode ocorrer através da adequação com os requisitos especificados. Isso deixa explícito que é necessário um ponto de partida para avaliar a qualidade de um produto. Quanto menor a conformidade dos sistemas com os requisitos especificados, menor qualidade ele terá.

Outro aspecto importante é o de atendimento às necessidades implícitas. Elas têm foco

na visão subjetiva do cliente com relação ao produto, tais como: aspectos de estética, itens de segurança, dentre outros (GUERRA; COLOMBO, 2009). Um exemplo de necessidade implícita para a compra de um carro seria ser confortável e ter um design sofisticado.

A qualidade de *software* pode ser dividida em qualidade do processo e qualidade do produto. De acordo com (SOMMERVILLE, 2011), processo de *software* é “um conjunto de atividades e resultados associados que levam à produção de um software. Esse processo pode envolver o desenvolvimento de software desde o início, embora, cada vez mais, ocorra o caso de um software novo ser desenvolvido mediante a expansão e a modificação de sistemas já desenvolvidos”. Assim, os estudos em qualidade do processo se referem à criação de normas e padrões que sistematizem e auxiliem o processo de desenvolvimento do *software*, a fim de gerar um produto com qualidade.

A qualidade do produto é baseada em normas que avaliam se o produto está atingindo os objetivos definidos. Para isso, é necessário especificar características desejáveis do software, que idealmente devem incluir valores quantitativos a serem alcançados. Quando um produto é medido com relação a essas características e valores é mais fácil determinar seu grau de qualidade (KOSCIANSKI; SOARES, 2006).

Uma das normas mais importantes a respeito de caracterização e medição de qualidade de produto de software é a norma ISO 25000. Essa norma é uma evolução das normas (ISO/IEC 9126, 2001) e (ISO/IEC 14598, 1999). Ela será apresentada na Seção 2.3.2 a seguir.

### **2.3.2 A Norma SQuaRE**

Em 1998, a Comissão de Estudos de Qualidade de *Software* da Associação Brasileira de Normas Técnicas (ABNT) percebeu que havia uma dificuldade em utilizar essas normas e, então, propôs a criação de um guia de utilização para auxiliar os usuários a compreenderem as ideias e conceitos das normas. A proposta foi tão bem aceita que uma versão com o idioma em inglês foi encomendada (KOSCIANSKI; SOARES, 2006).

Esse guia mostrou que era necessário repensar o modo como as normas eram apresentadas. O passo seguinte foi a reformulação dessas normas, o que deu origem ao projeto SQuaRE - *Software product Quality Requirements and Evaluation – the series of standards on product quality Requirements and Evaluation* (ISO/IEC 25000, 2005). O projeto SQuaRE fez uma reorganização das antigas normas, gerando uma nova divisão de assuntos que é apresentada na Figura 2.7 (KOSCIANSKI; SOARES, 2006).

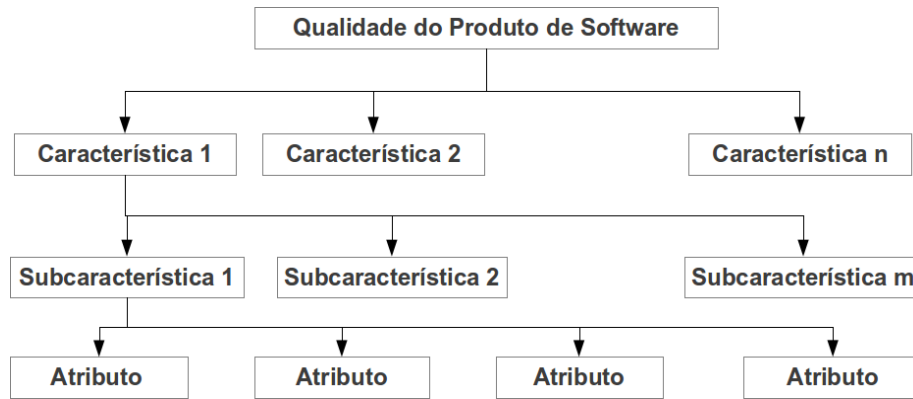
Divisão de Requisitos de Qualidade 2503n	Divisão de Modelo de Qualidade 2501n	Divisão de Avaliação de Qualidade 2504n
	Divisão de Gerenciamento de Qualidade 2500n	
	Divisão de Medição de Qualidade 2502n	

**Figura 2.7** - Divisão da norma SQuaRE, traduzida de (ISO/IEC 25000, 2005)

Cada divisão trata de um tópico e é composto de um conjunto de documentos, a seguir.

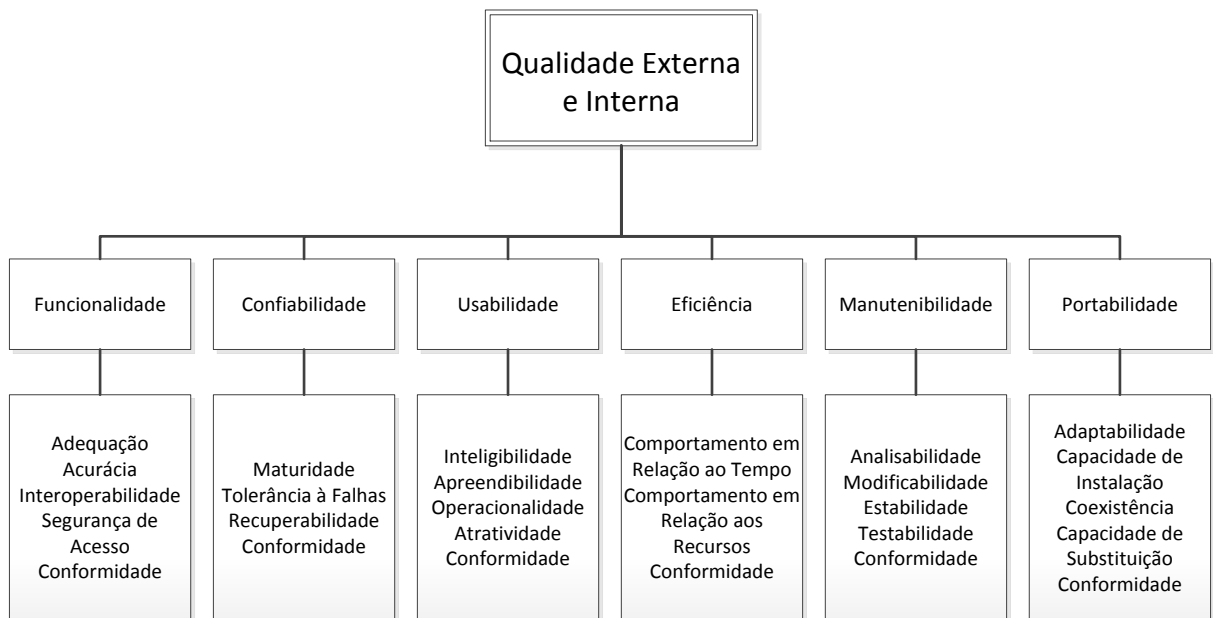
- Divisão de Gerenciamento da Qualidade, onde são definidos todos os termos que serão utilizados em todos os outros documentos e como está organizada a norma.
- Divisão do Modelo de Qualidade onde é apresentado o modelo de qualidade e guias práticos sobre como utilizá-lo.
- Divisão de Medição de Qualidade que inclui um modelo de referência para medições da qualidade de um produto, definições matemáticas de medidas e orientações práticas para sua aplicação.
- Divisão de Requisitos de Qualidade, cujos documentos que formam essa divisão ajudam na especificação de requisitos de qualidade.
- Divisão de Avaliação de Qualidade: essa divisão é composta por requisitos, recomendações e *guidelines* para avaliação de um produto de software.

O foco deste trabalho está nas divisões de modelo de qualidade e medição de qualidade. Na Figura 2.8 é apresentada a estrutura de um modelo de qualidade.



**Figura 2.8** - Estrutura de um Modelo de Qualidade, traduzida de (ISO/IEC 25000, 2005)

Segundo a norma, um modelo de qualidade é composto de características de qualidade, subcaracterísticas e atributos. Característica de qualidade é um conjunto de propriedades de um produto de software pela qual sua qualidade pode ser definida e avaliada (ISO/IEC 9126, 2001). Uma característica pode ser dividida em várias subcaracterísticas. Atributos são propriedades mensuráveis, físicas ou abstratas, de uma entidade. A norma ISO/IEC 9126 não define atributos no seu modelo de qualidade, pois estes variam de um produto para outro. Um exemplo de modelo de qualidade é apresentado na Figura 2.9.



**Figura 2.9** - Modelo de Qualidade da ISO/IEC 9126, traduzida de (MACHADO; SOUZA, 2004)

A característica Funcionalidade diz respeito à capacidade do produto de realizar as funções, as quais estão especificadas para fazer. Essa característica possui as seguintes subcaracterísticas: Adequação, Acurácia, Interoperabilidade, Segurança de Acesso e

Conformidade. Adequação é a capacidade de prover funções adequadas às necessidades do usuário. Acurácia é a precisão de um resultado. Interoperabilidade é a capacidade de se comunicar com outros programas. Segurança de acesso é a capacidade de proteger as informações contra uso indevido. Conformidade abrange padrões internos e legislações, essa subcaracterística está presente em todas as outras características.

Confiabilidade é a capacidade de o software manter um desempenho diante de certo contexto de uso. A subcaracterística Tolerância representa a capacidade de o software continuar operando diante de um acontecimento inesperado. Recuperabilidade é a capacidade do software voltar a funcionar depois do acontecimento de uma falha. Maturidade é a capacidade de evitar falhas e significa que quanto mais utilizado é o software, mais erros já foram encontrados e corrigidos e assim, a possibilidade de aparecer erros torna-se menor.

Usabilidade, definida na Seção 1.1, é dividida em quatro subcaracterísticas. Operacionalidade significa a capacidade do software ser controlado pelo usuário. Inteligibilidade é a capacidade de entendimento do usuário sobre a utilização do software. Apreendibilidade é a facilidade de aprendizado e, por fim, atratividade que é a capacidade do software atrair e manter a atenção do usuário. Outros componentes de usabilidade foram apresentados na Seção 2.2.1.

Eficiência é a capacidade do produto de software fornecer um desempenho apropriado, com relação aos recursos utilizados e sob um contexto de uso. Para essa característica foram estabelecidas duas subcaracterísticas: Comportamento em relação ao tempo e Comportamento em relação à recursos. O primeiro diz respeito à fatores que influenciam a velocidade de operação do software e a segunda engloba vários elementos de software que são utilizados na sua execução, como exemplo: espaço em disco, volume de tráfego em rede, carga de CPU.

Manutenibilidade é a facilidade de realizar modificações em um software. Uma modificação pode ser uma correção ou uma adaptação às mudanças de requisitos. A subcaracterística Modificabilidade refere-se a fatores que facilitam as mudanças no código. Analisabilidade é a facilidade com que se podem identificar deficiências e causas de falhas. Estabilidade é a capacidade de evitar que mudanças no produto levem ao surgimento de falhas. Por fim, a Testabilidade é a capacidade de que o software uma vez modificado seja validado.

Portabilidade é a capacidade do software operar em ambientes diferentes. A subcaracterística Capacidade de Instalação diz respeito à facilidade que um programa tem de ser instalado. Adaptabilidade é a capacidade do software ser instalado em diferentes ambientes sem necessitar de muitas ações de configuração. A Coexistência refere-se à capacidade de compartilhamento de recursos com outros produtos e por fim, a Capacidade de Substituição é a

capacidade de um programa substituir outro em um determinado ambiente.

O próximo passo para completar o modelo de qualidade é a definição de medidas. As normas definem medidas para cada uma das subcaracterísticas apresentadas no modelo, elas são utilizadas para indicar as características e/ou subcaracterísticas de qualidade em um produto de software. Para tanto, é necessário realizar medições de software. A próxima seção apresenta em mais detalhes conceitos sobre medição e medidas de software.

### 2.3.3 Medição e Medidas de Software

Segundo (VAN SOLINGEN; BERGHOUT, 1999) medição de software é “o processo contínuo de definição, coleta e análise de dados sobre o processo de desenvolvimento de software e seus produtos, a fim de entender e controlar o processo e seus produtos para fornecer informações significativas com o objetivo de melhorar esse processo e seus produtos”.

Como citado na Seção 1.1, o objetivo das medições é obter dados caracterizando um produto com relação ao modelo de qualidade. Quando medições são realizadas é possível identificar problemas, ineficiências e suas causas, o que leva a um melhoramento do produto que está sendo medido (PARK; GOETHERT; FLORAC, 1996).

A medição de software acontece através da aplicação de medidas de software. Uma medida, segundo a norma (ISO/IEC 25000, 2005), pode ser definida como uma variável ao qual é atribuído um valor como o resultado da medição. O termo “medidas” é usado pela norma (ISO/IEC 25000, 2005), no entanto, a norma ISO/IEC 9126 utiliza o termo métrica, na qual está relacionada ao método e escala de medições definidos.

Uma escala de medições fornece valores e unidades para a descrição dos atributos de entidades (RAMOS, 2004). Elas podem dos tipos descritos a seguir.

- Nominal, que consiste em atribuir uma classificação para os atributos medidos, pode ser um nome ou um valor. Exemplo: estado civil (solteiro, casado, divorciado, viúvo);
- Ordinal, que consiste em atribuir ordem aos atributos medidos. Exemplo: falha em software por grau de severidade (negligenciado, marginal, crítico e catastrófico);
- Intervalo, que considera a ordem como na escala ordinal, mas também consiste em medir a diferença entre os pontos da escala. Exemplo: temperatura, horário; e
- Taxa, é semelhante à escala de intervalo, no entanto ela possui o marco zero (indicador da falta do atributo).

Uma medida de software pode ser classificada quanto ao seu tipo (ISO/IEC 25000, 2005):

- Tamanho, por exemplo, tamanho da função, tamanho da fonte;
- Tempo, por exemplo, o tempo transcorrido, o tempo do usuário;
- Contagem, por exemplo, número de mudanças, número de falhas; e
- Fonte de entrada para medição é a fonte de dados utilizados na medição, por exemplo, um documento de requisitos ou um diagrama de arquitetura;

Outras informações são utilizadas para documentar uma medida de software, a forma como as normas as documentam são apresentadas na Seção 2.3.4 a seguir.

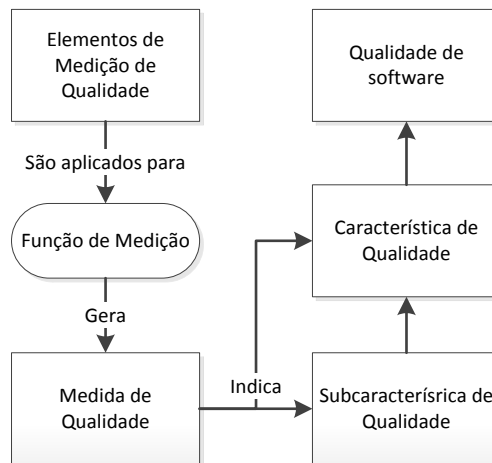
### 2.3.4 Definição de Medidas nas Normas

A ISO/IEC 9126 propõe que as medidas devem ser organizadas da seguinte forma.

- Nome, que é a identificação da métrica (*e.g.*, facilidade de aprender funções);
- Objetivo da métrica, o qual é expresso através de uma pergunta a ser respondida através da aplicação da métrica (*e.g.*, quanto tempo para o usuário aprender uma função?);
- Método de aplicação, que provê um esboço da aplicação da métrica (*e.g.*, conduzir um teste com o usuário e observar o seu comportamento);
- Medição, fórmula e cálculo dos elementos de dados, os quais fornecem uma fórmula de medição e explica o significado dos elementos de dados utilizados (*e.g.*,  $t$ =tempo médio gasto para que o usuário aprenda a utilizar uma função corretamente);
- Interpretação dos valores medidos, que provê um intervalo e valores apropriados (*e.g.*,  $T > 0$ . Quanto menor o valor de  $T$  melhor);
- Tipo de escala da métrica (*e.g.*, taxa);
- Tipo de medida (*e.g.*,  $t$ =tempo);
- Referência da ISO/IEC 12207, que identifica o processo do ciclo de vida em que a métrica é aplicada (*e.g.*, validação e operação); e
- Alvo, que identifica os usuários dos resultados das medições (*e.g.*, usuários e mantenedor).

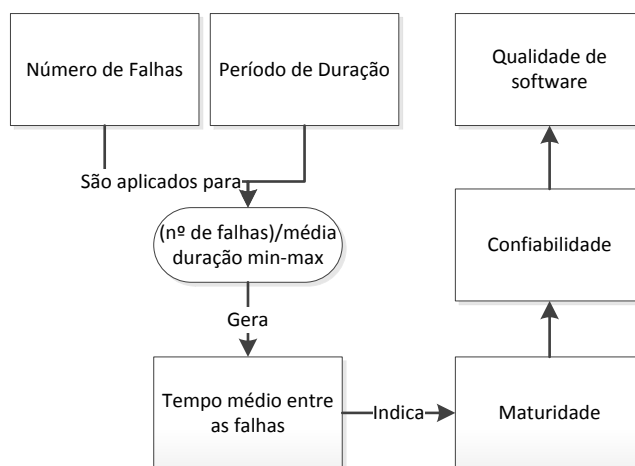


O termo métrica foi utilizado anteriormente, pois, como citado na seção 2.3.3, a norma ISO 9126 utiliza esse termo, no entanto, a norma SQuaRE não o utiliza mais. Nela um novo conceito em relação a medições foi criado: Elementos de Medição de Qualidade (EMQ). Esses elementos correspondem a uma medida definida em termos de uma propriedade a ser quantificada e um método de medição para quantificá-la. A Figura 2.10 apresenta o relacionamento entre EMQ, medidas de qualidade, função de medição e o modelo de qualidade.



**Figura 2.10** - Relacionamento entre EMQ, função de medição, medida de qualidade e modelo de qualidade, traduzida de (ISO/IEC 25000, 2005)

Os EMQs são aplicados para formar uma função de medição, que por sua vez gera a medida de qualidade. Segue um exemplo na figura abaixo:



**Figura 2.11** - Exemplo de elementos de qualidade, função de medição e medida de qualidade, traduzida de (ISO/IEC 25000, 2005)

Nesse exemplo, os elementos de medição de qualidade são o número de falhas e o período de duração. Cada um desses EMQs possui uma propriedade a ser quantificada e um método de medição para quantificá-la. Na Figura 2.11, o elemento de qualidade de medição

“número de falhas” possui “falhas” como propriedade a quantificar e testes de software como um método de medição, que é um método que deverá atribuir um valor numérico a falha. A combinação entre números de falhas e o período de duração formam a medida chamada Tempo Médio entre Falhas.

Para registrar esses aspectos, a norma SQuaRE define um padrão de documentação das medidas de qualidade, descrito a seguir.

- ID: código de identificação para auxiliar na referência da medida;
- Nome: a medida de qualidade deve ter um nome único;
- Descrição: a medida deverá ser expressa como uma questão a ser respondida pela aplicação da medida;
- Função de medição e EMQs;

Alguns outros exemplos de medidas são apresentados nas tabelas abaixo:

**Tabela 2.1 - Medidas para Subcaracterística Compreensibilidade, traduzida de (ISO/IEC 9126, 2001)**

Nome	Propósito	Método	Fórmula	Interpretação
Funções Evidentes	Qual a proporção de funções do produto é evidente para o usuário?	Contar o nº de funções que são evidentes para os usuários e comparar com o nº total de funções	$X=A/B$ A=nº de funções evidentes para o usuário B=nº total de funções	$0 \leq X \leq 1$ Quanto mais perto de um, melhor.
Compreensibilidade de funções	Qual a proporção de funções do produto o usuário será capaz de entender corretamente?	Contar o nº de funções na interface do usuário que os objetivos são compreendidos por um usuário e comparar com o nº de funções existentes	$X=A/B$ A= o nº de funções na interface do usuário que os objetivos são compreendidos B=nº total de funções existentes	$0 \leq X \leq 1$ Quanto mais perto de um, melhor.

**Tabela 2.2** - Medidas para a Subcaracterística Estabilidade, traduzida de (ISO/IEC 9126, 2001)

Nome	Propósito	Método	Fórmula	Interpretação
Impacto da mudança	Qual é a frequência dos impactos adversos após a modificação?	Contar o número dos impactos adversos detectados depois da modificação e compará-lo com o número de modificações executadas	$X=1-A/B$ A=nº de impactos adversos detectados B=nº de modificações feitas	$0 \leq X \leq 1$ Quanto mais perto de um, melhor
Localização do impacto da modificação	Quão grande é o impacto da modificação no produto de software?	Contar o nº de variáveis afetadas pela modificação e compará-lo ao total de nº de variáveis no produto	$X=A/B$ A= nº de dados de variáveis afetadas pela modificação B=nº total de variáveis	$0 \leq X \leq 1$ Quanto mais perto de 0, menor é o impacto das modificações

Conforme apresentado anteriormente, existem várias medidas de software para avaliação de diferentes características. Para que o processo de medição de software seja eficaz, é necessário que ele esteja alinhado com os objetivos que se deseja alcançar, portanto, definir um conjunto adequado de medidas de software torna-se uma atividade essencial (RAMOS, 2004). Para tanto, existem abordagens de definição de medidas que serão apresentadas na Seção 2.3.5 seguir.

### 2.3.5 Abordagem para Definição de Medidas

Na literatura existem várias abordagens para definição de medidas de *software*: *Practical Software Measurement* (PSM) (HUGHES, 2000), Guia de Medição Orientada a Objetivos (GQ(I)M) (PARK; GOETHERT; FLORAC, 1996) e o *Goal-Question-Metric* (GQM) (BASILI; ROMBACH, 1994).

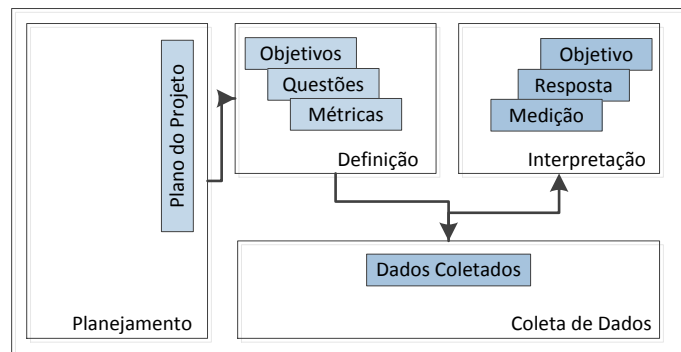
O PSM é um modelo para estruturação da mensuração em um projeto de software. Essa abordagem busca resolver o problema de como especificar formalmente medidas de *software* e como aplicá-las em um processo de medição (HUGHES, 2000).

A abordagem (GQ(I)M) surgiu devido a dificuldade em se derivar os objetivos de medição no GQM, então, para auxiliar nessa atividade é indicado utilizar o GQ(I)M. Este guia é composto por dez passos: (i) identificar objetivos de negócio, (ii) identificar o que se quer saber ou aprender, (iii) identificar subobjetivos, (iv) identificar entidades e atributos relacionados, (v) formalizar os objetivos de medição, (vi) identificar questões quantificáveis e indicadores, (vii)

identificar elementos de dados que serão coletados, v(iii) definir medidas e torná-las operacionais, (ix) identificar ações a serem tomadas para as medições e (x) preparar um plano de medição.

O paradigma GQM, que é o utilizado nessa dissertação, é uma abordagem orientada a objetivos para a mensuração de produtos e processos de *software*. Essa abordagem é baseada na premissa de que as medições de *software* devem ser orientadas aos objetivos de medição, que por sua vez devem ser orientados aos objetivos da organização (BASILI; ROMBACH, 1994).

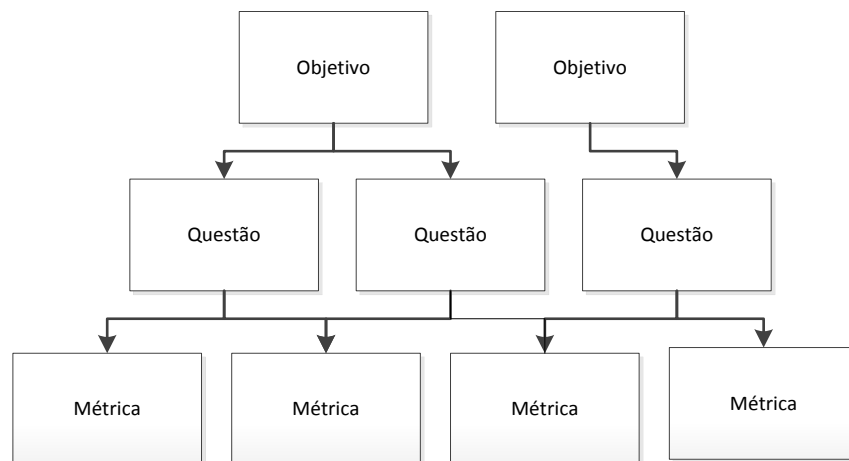
O GQM consiste de quatro fases (Figura 2.12): planejamento, definição, coleta de dados e interpretação. Essas fases são apresentadas a seguir.



**Figura 2.12** - Fases do Paradigma GQM, traduzida de (VAN SOLINGEN; BERGHOUT, 1999)

**Planejamento:** os objetivos dessa fase são estabelecer um time GQM (time independente do projeto que se pretender realizar as medições), selecionar uma área para melhoria (por exemplo, nível de satisfação dos clientes), selecionar um projeto de aplicação e o time do projeto, criar o plano de projeto e treinar todos os participantes do projeto.

**Definição:** nesta fase são definidos os objetivos, as questões correspondentes aos objetivos e as métricas que respondem a essas questões, estes se apresentam de maneira hierárquica, como ilustrado na Figura 2.13.



**Figura 2.13** - Estrutura do Paradigma GQM, traduzida de (VAN SOLINGEN; BERGHOUT, 1999)

Como pode ser visto na Figura 2.13, uma métrica pode ser utilizada para responder diferentes questões de diferentes objetivos. Tais objetivos de medição devem refletir o que se pretende alcançar com a medição, eles devem ser bem definidos e claros, para isso, são utilizados *templates* para apoiar a definição deles. Um *template* é ilustrado na Tabela 2.3.

**Tabela 2.3** - Template para Definir Objetivos, traduzida de (VAN SOLINGEN; BERGHOUT, 1999)

<b>Analisar</b>	O objeto sob medição
<b>Com o propósito de</b>	Entender, controlar, melhorar o objeto
<b>Com respeito à</b>	Foco de qualidade do objeto que a medição foca
<b>Ponto de vista</b>	As pessoas que medem o objeto
<b>No contexto</b>	Ambiente em que a medição ocorre

Dado que os objetivos estão definidos, um conjunto de questões é utilizado para tentar caracterizar o objeto de medição (i.e., produto, processo, recurso) com relação a um objetivo específico, em seguida, um conjunto de métricas é associado a todas as questões, a fim de responder de forma quantitativa.

Para definir tais informações, entrevistas são realizadas com os especialistas no produto que se deseja avaliar. O objetivo das entrevistas é coletar as informações importantes sobre o produto que servirão como entrada para definir as questões e as medidas. O principal objetivo dessas entrevistas é tornar o conhecimento implícito em explícito. Para apoiar a comunicação entre os profissionais do GQM e os especialistas do produto, o time do GQM utiliza o que se chama “*abstraction sheets*” (Figura 2.14), um tipo de *template* que provê uma abordagem estruturada para focar em questões relevantes.

<b>Foco de qualidade</b>	<b>Fatores de variação</b>
Quais são as possíveis métricas para medir o objeto do objetivo de acordo com os especialistas?	Quais fatores os especialistas acham que influenciam o resultado dessa métrica?
<b>Hipóteses de <i>baseline</i></b>	<b>Impacto nas hipóteses de <i>baseline</i></b>
Qual é o conhecimento atual dos especialistas com respeito a essas métricas? Suas expectativas são documentadas como hipótese de <i>baseline</i> para as métricas.	Como esses fatores influenciam as medições atuais? Que tipo de dependências há entre as métricas e os fatores de variação?

**Figura 2.14** - *Abstraction Sheets*, traduzida de (VAN SOLINGEN; BERGHOUT, 1999)

Os dados obtidos a partir dessas entrevistas apoiam a definição de questões e medidas para alcançar o objetivo definido.

**Coleta de dados:** nesta fase são definidos todos os procedimentos para coleta de dados e por fim, os dados são coletados. Tais dados podem ser coletadas de forma manual, através de preenchimento de formulários e/ou por forma automatizada, através de funções e/ou programas. Essa fase corresponde à execução dos estudos de casos.

**Interpretação:** essa fase inclui a análise dos dados coletados, a preparação do material para apresentação e relatório dos resultados encontrados.

## 2.4 Considerações Finais

Neste capítulo foram apresentados os conceitos das áreas que investigamos nesse trabalho: Sistemas Ubíquos, Interação Humano-Computador e Qualidade de Software.

Com relação à área dos sistemas ubíquos, definições, características e exemplos foram apresentados a fim de diferenciar tais sistemas dos sistemas tradicionais. Algumas características que evidenciam tais diferenças são a sensibilidade ao contexto, a mínima intervenção do usuário, a heterogeneidade de dispositivos e serviços e a onipresença dos serviços.

Como sensibilidade ao contexto é uma das principais características, dado que é impossível conceber um sistema ubíquo sem ser sensível ao contexto, essa característica foi mais aprofundada ao apresentar conceitos importantes que serão considerados no momento de definir as medidas de software propostas neste trabalho.

Com relação a IHC, alguns conceitos que compõem essa área foram apresentados, como os critérios de qualidade do uso de sistemas e os métodos de avaliação comumente utilizados para avaliar a qualidade da interação com os sistemas. Nesta seção também foi abordado como é tratada a IHC nos sistemas ubíquos, ela prevê que a interação com esse tipo de sistema seja implícita, assim o usuário não é necessariamente destinado a interagir explicitamente com um sistema computacional.

Por fim, foram apresentados conceitos sobre Qualidade de *Software*, mais especificamente sobre qualidade do produto. Definições sobre o que são modelos de qualidade, características de qualidade, medidas de software e abordagens existentes para definir e selecionar medidas de software apropriadas para uma avaliação da qualidade foram apresentadas.

A fundamentação teórica apresentada nesse capítulo fornece o embasamento necessário para o desenvolvimento deste trabalho, cujo objetivo é definir medidas de software para avaliar

características de qualidade de interação humano-computador em sistemas ubíquos. O próximo capítulo apresenta uma investigação para definir tais características de qualidade.

# 3 Investigação e Definição de Características de Qualidade para IHC em Sistemas Ubíquos

Este capítulo apresenta um estudo para identificar características de qualidade de interação humano-computador em sistemas ubíquos. Esse estudo utiliza como metodologia de busca o Mapeamento Sistemático da literatura. Como resultado desse mapeamento, uma lista de características pertinentes para uma avaliação da qualidade é definida. Para apresentar este resultado, o capítulo está organizado da seguinte forma: a Seção 3.1 apresenta uma explanação sobre Mapeamento Sistemático (MS), a Seção 3.2 apresenta o planejamento realizado para executar um MS para esse trabalho, a Seção 3.3 apresenta a execução do MS, a Seção 3.4 apresenta os resultados encontrados pelo MS. A Seção 3.5 apresenta a lista de características definidas a partir desse MS e, por fim, a Seção 3.6 apresenta as considerações finais sobre o capítulo.

## 3.1 Metodologia de Busca: Mapeamento Sistemático

O Mapeamento Sistemático (MS) é uma metodologia que provê uma visão geral de uma área de pesquisa, identificando evidências, indicando a quantidade e apresentando os resultados encontrados em forma de um mapa sistemático (PETERSEN et al., 2008). Essa metodologia é um tipo de Revisão Sistemática (RS), que é um método para identificar, avaliar e interpretar todos os estudos relevantes para uma questão de pesquisa particular, tópico de pesquisa ou fenômeno de interesse (KITCHENHAM; CHARTERS, 2007). RS são chamadas de estudos secundários, pois agregam resultados de estudos primários, os quais podem ser experimentos controlados, estudos de casos, *survey* ou *action-research*.

Diferentemente de revisões informais da literatura, uma RS é composta de três fases bem definidas: planejamento, execução e resultados. Na fase de planejamento, encontra-se uma atividade que diferencia revisões sistemáticas de revisões informais: a definição de um documento chamado protocolo de pesquisa. Esse documento especifica as questões de pesquisa que serão endereçadas, as fontes de pesquisa, a string de busca, critérios de seleção dos estudos e estratégia



de extração de dados. Assim, essa metodologia possibilita que os leitores possam acessar seu rigor e completude, possibilitando também sua repetitividade. A fase de execução é a fase na qual a busca pelos estudos é iniciada e os dados dos estudos encontrados e selecionados são extraídos e, por fim, a fase de resultados, no qual os dados são sumarizados e divulgados (KITCHENHAM; CHARTERS, 2007).

Existem várias vantagens em utilizar uma metodologia sistemática na busca de trabalhos de uma determinada área de pesquisa: sumarizar evidências existentes sobre um fenômeno ou tecnologia; identificar lacunas em pesquisas atuais a fim de sugerir trabalhos para pesquisas futuras; e prover uma estrutura para a atividade de pesquisa. Além disso, RS também pode ser utilizada para examinar se evidências comprovam ou contradizem hipóteses teóricas, e até mesmo para apoiar a geração de novas hipóteses de pesquisa (KITCHENHAM; CHARTERS, 2007).

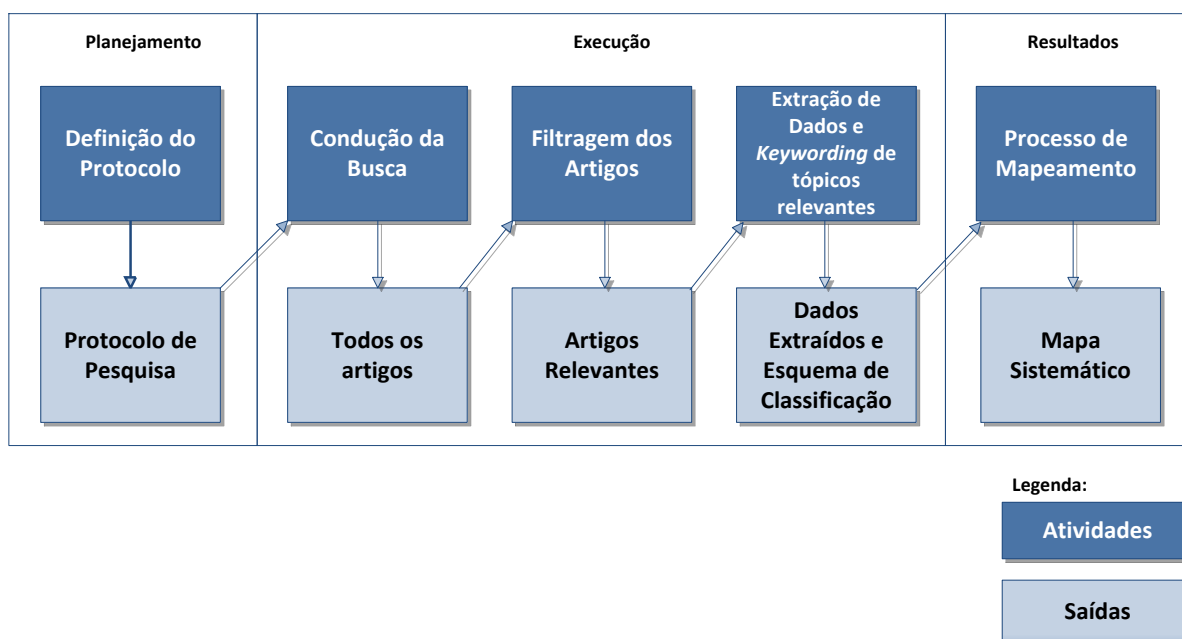
O MS utiliza a mesma base da RS, no entanto, essas metodologias possuem algumas diferenças, como (KITCHENHAM; CHARTERS, 2007):

- Estudos de mapeamento têm, geralmente, questões de pesquisa mais amplas, geralmente relacionadas com tendências e tópicos de pesquisa;
- Os termos de pesquisa para estudos de mapeamento são mais abrangentes do que para revisões sistemáticas, pois seu objetivo é ampla cobertura ao invés de foco;
- O processo de extração de dados para MS também é muito mais amplo do que o processo de extração de dados de RS e pode ser chamado de uma fase de classificação ou categorização; e
- A fase de análise de dados do MS objetiva sumarizar dados e classificá-los. A divulgação é feita através de um mapa sistemático, que é um gráfico de bolhas, onde o tamanho da bolha é proporcional ao número de estudos que correspondem ao par de categorias x-y do gráfico, o que indica a quantidade de evidências encontradas.

Segundo (KITCHENHAM; CHARTERS, 2007), um MS é muito útil para estudantes que necessitam conhecer uma visão geral de uma área ou tópico de pesquisa. Assim, o estudo de mapeamento sistemático foi o escolhido para realizar o objetivo específico 1 desse trabalho.

O tópico de pesquisa endereçado para o MS realizado para este trabalho foi a identificação de características de qualidade, medidas de *software* e modelos de qualidade relacionados à avaliação da IHC em sistemas ubíquos. O processo seguido para realizar esse estudo foi adaptado de (NETO et al., 2011) e (PETERSEN et al., 2008), representado na Figura

3.1, e é composto de três principais fases: Planejamento, Execução e Resultados. Essas três fases são detalhadas nas seções a seguir.



**Figura 3.1** - Processo de MS, adaptado de (NETO et al., 2011)(PETERSEN et al., 2008)

## 3.2 Planejamento

A fase de planejamento, como já mencionado anteriormente, contém a principal atividade que diferencia uma RS da revisão de literatura convencional. Essa fase possui a atividade de definição do protocolo de pesquisa. No MS, essa não é uma atividade obrigatória, entretanto, ela foi incluída nesse MS por ser um meio de planejar e documentar todo o processo de pesquisa. O protocolo é um documento composto das seguintes informações: questões de pesquisa, *string* de busca, fontes de pesquisa e critérios de seleção. A seguir são apresentadas cada uma dessas informações para este estudo:

- Questões de Pesquisa (QP) – O objetivo do estudo é identificar quais são as características que influenciam a qualidade da IHC em sistemas ubíquos e quais são as medidas existentes na literatura, assim as questões QP1 e QP2 foram definidas. Sabendo ainda que características e medidas geralmente são organizadas em modelos de qualidade definiu-se, então, a QP3. A seguir, estão descritas as questões de pesquisa estabelecidas:
  - QP 1 – Quais características de qualidade são utilizadas para avaliações de IHC em sistemas ubíquos?
  - QP 2 – Quais são as medidas de qualidade utilizadas para a avaliação de IHC

nesses sistemas?

c) QP 3 – Existem modelos de qualidade para avaliação da IHC em Sistemas Ubíquos?

- Termos chaves – Os termos chave foram derivados das questões de pesquisa. Primeiro foram identificados os objetos de interesse (características, medidas e modelos de qualidade), depois o propósito (avaliação de IHC) e, por fim, o contexto (sistemas ubíquos). Após isso, alguns sinônimos e palavras alternativas foram adicionados no conjunto de termos chave (Tabela 3.1).

**Tabela 3.1 - Termos chave**

<b>Objeto</b>	<b>Propósito</b>	<b>Contexto</b>
	<u>HCI evaluation</u>	
	<u>Human-Computer Interaction Evaluation</u>	
	<u>Human-Computer Interface Evaluation</u>	
	<u>User Interface Evaluation</u>	
<u>Characteristic</u>	<u>Interaction Evaluation</u>	
<u>Measure</u>	<u>Usability Evaluation</u>	<u>Ubiquitous</u>
<u>Metric</u>	<u>HCI Assessment</u>	<u>Pervasive</u>
<u>Quality Model</u>	<u>Human-computer Interaction Assessment</u>	
<u>Framework</u>	<u>Human-computer Interface Assessment</u>	
	<u>User Interface Assessment</u>	
	<u>Interaction Assessment</u>	
	<u>Usability Assessment</u>	

A palavra “usabilidade” foi considerada como sinônimo da palavra avaliar IHC pelo fato de que a maioria dos estudos sobre avaliação de IHC refere-se como avaliação de usabilidade, sendo quase um termo intercambiável.

Outro detalhe que vale ressaltar é que o uso dos termos “sistemas” e/ou “aplicações” foi evitado, pois de acordo com (PETERSEN et al., 2008), isso é uma restrição e o MS objetiva ter uma visão geral de uma área de pesquisa. Se forem considerados somente certos tipos de estudo a visão geral pode ser tendenciosa e o mapa pode não ficar completo.

- String de Busca – A *string* de busca foi a combinação dos termos chave apresentados anteriormente:

((*characteristic* OR *measure* OR *metric* OR “*quality model*” OR *framework*) AND (HCI OR

*“human-computer interaction” OR “human-computer interface” OR “user interface” OR interaction OR usability) AND (evaluation OR assessment) AND (ubiquitous OR pervasive)*

- Fontes de Pesquisa – Foram escolhidas seis bibliotecas digitais: *ACM Digital Library* (<http://dl.acm.org/>), *IEEE Xplore* ([ieeexplore.ieee.org/](http://ieeexplore.ieee.org/)), *Scopus* ([www.scopus.com/scopus/home.url](http://www.scopus.com/scopus/home.url)), *Science Direct* (<http://www.sciencedirect.com/>), *SpringerLink* ([www.springer.com/](http://www.springer.com/)) and *Compendex* ([www.engineeringvillage.com/](http://www.engineeringvillage.com/)).
- Cr terios de Sele o – Foram definidos cinco cr terios de sele o a fim de filtrar os estudos mais apropriados:
  - a) O estudo deve ser escrito em ingl s;
  - b) O estudo deve ser datado a partir de 1991, pois o termo “computa o ub qua” foi criado neste ano;
  - c) O estudo deve estar dispon vel na internet;
  - d) O estudo deve apresentar iniciativas relacionadas com a avalia o de IHC em sistemas ub quos e n o em outros contextos, como desktop ou web; e
  - e) O mesmo estudo que foi publicado em diferentes artigos, somente o mais completo e recente ser  includo.

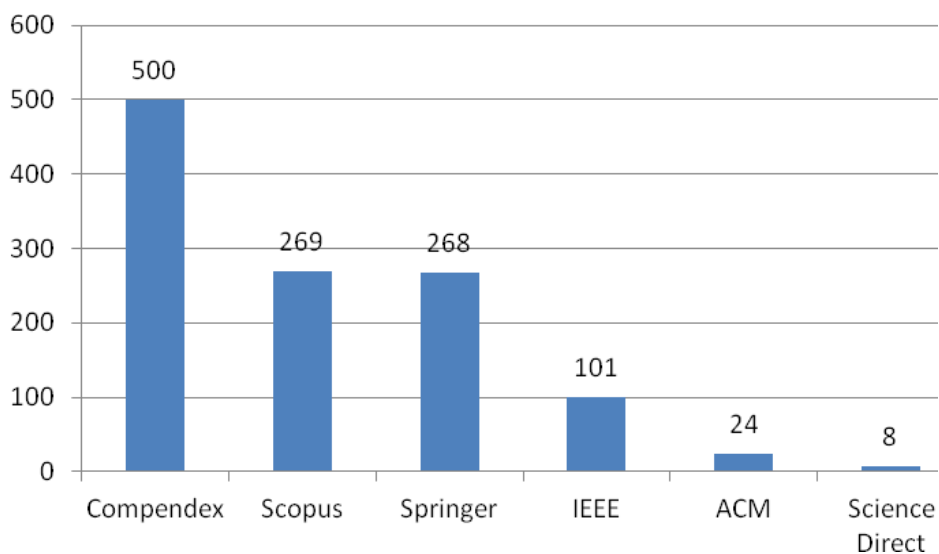
## 3.3 Execu o

### 3.3.1 Condu o da Busca

A primeira atividade da fase Execu o (Condu o da busca) representa a sele o dos artigos. Essa sele o   feita atrav s da aplica o da string de busca dentro das m quinas de busca das fontes de pesquisa (*ACM, IEEE, Scopus, Springer, Science Direct e Compendex*). Feito isso, todas as refer ncias dos artigos, incluindo os abstracts foram importadas para a ferramenta Start<sup>3</sup> (Software livre para apoio de atividades de revis o sistem tica). Foram obtidos 1170 artigos (Figura 3.2), 500 da Compedex (42,7%), 269 da Scopus (23%), 268 da Springer (22,9%), 101 da IEEE (8,6%), 24 da ACM (2,1%), 8 da Science Direct (0,7%)

---

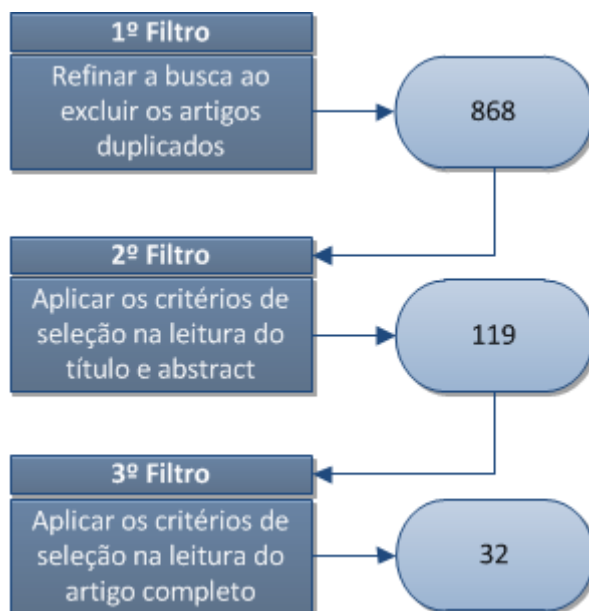
<sup>3</sup> <http://lapes.dc.ufscar.br/software/start-tool>



**Figura 3.2** – Quantidade de Estudos vs Fonte de Pesquisa

### 3.3.2 Filtragem

A segunda atividade da fase Execução (Filtragem) consiste na utilização de três filtros para selecionar os artigos mais apropriados, tais filtros estão descritos na Figura 3.3.



**Figura 3.3** - Filtros

O primeiro filtro consiste de excluir os artigos duplicados, pois alguns artigos apareceram em várias fontes, assim, somente um deles foi incluído. No total, foram identificados 302 artigos duplicados (26%). Assim, 868 artigos (74%) passaram para o segundo filtro.

O segundo filtro consistiu de aplicar os critérios de seleção definidos através da leitura do título e do abstract do artigo. Essa análise foi realizada com o apoio de outro pesquisador, que revisava a seleção do outro, em caso de dúvidas. No total, foram rejeitados 749 artigos (86,30%) e aceitos 119 (13,70%), os quais passaram para o terceiro filtro.

O terceiro filtro consistiu na leitura completa dos 119 artigos, para tanto, os artigos foram baixados, uma leitura detalhada foi realizada e os critérios de seleção mais uma vez foram aplicados. Essa atividade foi executada com o auxílio de três pesquisadores. Dois outros pesquisadores participaram da revisão em alguns artigos que causaram dúvidas. Como resultado, 87 artigos foram rejeitados (73%), dos quais 85 foram rejeitados pelo critério de seleção d) e 2 artigos pelo critério e), descritos na Seção 3.2.1, e 32 artigos foram aceitos.

Um resumo dessa análise para todas as fontes está apresentado na Tabela 3.2. A Figura 3.4 apresenta a quantidade dos artigos finais selecionados pelo ano de publicação.

**Tabela 3.2 - Sumário de artigos selecionados por fonte e filtro**

Fonte	Inclusão	1º Filtro	2º Filtro	3º Filtro
Compedex	500	282	36	6
Scopus	269	241	52	16
Springer	268	265	18	7
IEEE	101	63	12	2
ACM	24	15	1	1
Science Direct	8	2	0	0
<b>Total</b>	<b>1170</b>	<b>868</b>	<b>119</b>	<b>32</b>



**Figura 3.4 - Ano de Publicação**

### 3.3.3 *Keywording* de Tópicos Relevantes e Extração de Dados

Para finalizar a fase da Execução, os dados devem ser extraídos para que as questões de pesquisa sejam respondidas e para que o mapa sistemático seja criado. Na criação do mapa, é necessário definir facetas que correspondem aos eixos x-y do gráfico de bolhas, essas facetas são os esquemas de classificação. Elas são definidas a partir do *Keywording* de tópicos relevantes. Essa atividade consiste na busca por palavras chaves e conceitos que refletem a contribuição do artigo.

Durante a leitura, foi possível identificar os principais conceitos: definição do modelo de qualidade, *frameworks* conceituais, medidas e características de qualidade. Tais conceitos foram definidos como uma das facetas para a construção do gráfico de bolhas (mapa sistemático) e são apresentados na Tabela 3.3.

**Tabela 3.3 - Faceta Tipo de Contribuição**

<b>Categoria</b>	<b>Descrição</b>
Modelo de Qualidade	O artigo apresenta um modelo semelhante ao definido pela norma SQuaRE (ISO/IEC 25000, 2005): características, subcaracterísticas e medidas em um modo hierárquico. Neste caso, as medidas são apresentadas.
Framework de Características	O artigo organiza as características e/ou subcaracterísticas como uma lista de fatores que devem ser avaliados em um sistema ubíquo. Nenhuma medida é apresentada.
Framework de Medidas	O artigo apresenta uma lista de medidas organizadas dentro de características de qualidade.
Questões de Qualidade	O artigo apresenta características que podem ser consideradas em uma avaliação de sistemas ubíquos, mas não estão explicitamente definidas como características de qualidade no artigo.

A faceta apresentada na Tabela 3.3 representa qual nível de contribuição foi dado pelo artigo. Dessa forma, é possível classificar o artigo em uma dessas categorias de contribuição.

Para formar o segundo eixo do mapa, a faceta utilizada foi a definida por (WIERINGA et al., 2005) e utilizada por (PETERSEN et al., 2008), apresentada na Tabela 3.4. Nessa faceta, o artigo também deve ser classificado em uma das categorias, isso significa que ela é excludente, assim como a faceta tipo de contribuição.

**Tabela 3.4** - Faceta Tipo de Pesquisa, traduzida de (PETERSEN et al., 2008)

<b>Categoria</b>	<b>Descrição</b>
Pesquisa de Validação	Quando as técnicas investigadas são novas e ainda não têm sido implementadas na prática, mas somente em experimentos controlados.
Pesquisa de Avaliação	Quando as técnicas são implementadas na prática.
Proposta de Solução	Quando uma solução para um problema é proposta e ela é apresentada por um pequeno exemplo ou alguma argumentação.
Artigos Filosóficos	Quando o artigo esboça um novo olhar para trabalhos existentes através de uma estruturação do campo em forma de taxonomia ou framework conceitual.
Artigos de Opinião	Quando artigos expressam a opinião pessoal.
Artigos de Experiência	Quando explicam o quê e como algo tem sido feito na prática.

Finalmente, a extração de dados foi realizada para cada um dos 32 artigos finais selecionados. Para esse fim, a Tabela 3.5 foi utilizada para guiar essa atividade. Nela, as informações necessárias para responder cada questão de pesquisa foram extraídas, o que significa responder se o artigo define características de qualidade (respondendo a QP 1), se o artigo define medidas de software (respondendo a QP 2) e se o artigo define um modelo de qualidade (respondendo a QP 3). Além disso, foi necessário classificar o artigo considerando o esquema de classificação definido (Tabela 3.3 e Tabela 3.4).

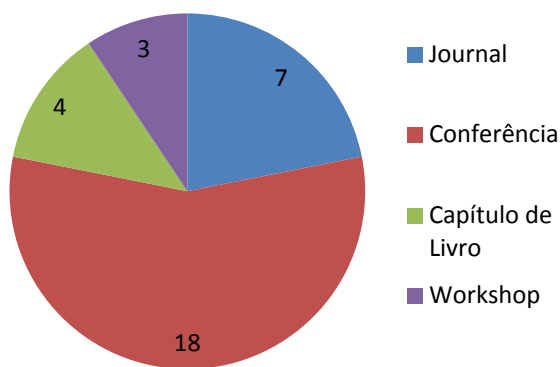
**Tabela 3.5** - Formulário de Extração de Dados

Modelo de Qualidade	<input type="checkbox"/> Sim <input type="checkbox"/> Não
Característica de Qualidade	
Medidas de Software	
Domínio da Aplicação	
Tipo de Publicação	<input type="checkbox"/> Livro <input type="checkbox"/> Capítulo de Livro <input type="checkbox"/> Conferência <input type="checkbox"/> <i>Journal</i> <input type="checkbox"/> <i>Workshop</i>
Tipo de Pesquisa	<input type="checkbox"/> Pesquisa de Validação <input type="checkbox"/> Pesquisa de Avaliação <input type="checkbox"/> Proposta de Solução <input type="checkbox"/> Artigos Filosóficos <input type="checkbox"/> Artigos de Opinião <input type="checkbox"/> Artigos de Experiência
Tipo de Contribuição	<input type="checkbox"/> Modelo de Qualidade <input type="checkbox"/> <i>Framework</i> de Características <input type="checkbox"/> <i>Framework</i> de Medidas <input type="checkbox"/> Questões de Qualidade

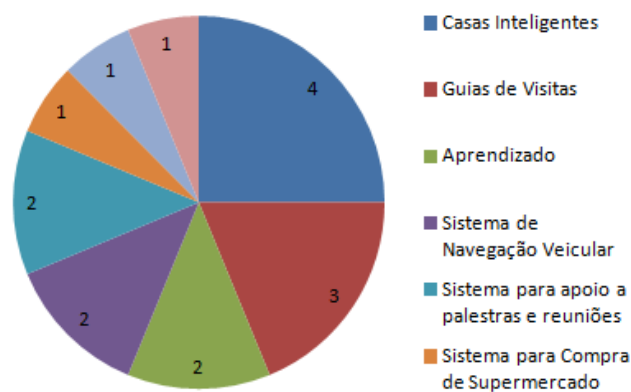


### 3.4 Resultados

A última fase do mapeamento é apresentar os resultados, respondendo as questões de pesquisa e apresentando o mapa sistemático. Esse MS selecionou ao final dos filtros 32 artigos: (ABI-CHAR et al., 2010; CAPPIELLO; PUGLIA; VITALETTI, 2009; CHANG; LIN, 2011; DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011; DE MOOR et al., 2010; EVERS et al., 2010; HAAPALAINEN et al., 2010; IQBAL et al., 2005; JAFARI et al., 2010; JIA; COLLINS; NIXON, 2009; KEMP; JOHNSON; THOMPSON, 2008; KIM; CHOI; JI, 2008; KO; KOO; JIMENEZ-MOLINA, 2010; KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008; KRYVINSKA; STRAUSS; ZINTERHOF, 2011; LEE; YUN, 2012; LEE et al., 2008; LIAMPOTIS et al., 2009; RANGANATHAN et al., 2005; ROSS; BURNETT, 2001; RUBIO; BOZO, 2007; SCHALKWYK et al., 2010; SCHOLTZ; CONSOLVO, 2004; SOUSA; PENTIKOUSIS; CURADO, 2011; SUN; DENKO, 2008; THOMPSON; AZVINE, 2004; TOCH, 2011; WAGNER; TOFTEGAARD; BERTELSEN, 2012; WAIBEL et al., 2010; WARNOCK, 2011; WEIHONG-GUO et al., 2008; WU; FU, 2012; ZHANG et al., 2006). A Figura 3.5 apresenta a distribuição de artigo por tipo de publicação (livro, conferência, *workshop*, *journal* e capítulos de livro).



**Figura 3.5** - Distribuição de Artigos por Tipo de Publicação



**Figura 3.6** - Domínios de Aplicações Ubíquas

A Figura 3.6 apresenta os domínios de aplicações ubíquas utilizadas nos artigos encontrados nesse MS. A maioria dos artigos utilizaram serviços para Casas Inteligentes como exemplo de aplicação em seus trabalhos (WU; FU, 2012)(LEE et al., 2008)(LEE; YUN, 2012) (KO; KOO; JIMENEZ-MOLINA, 2010) e, em seguida, estão os guias de visitas, como domínio de aplicação mais utilizado nos trabalhos encontrados por esse MS (SCHOLTZ; CONSOLVO, 2004)(CHANG; LIN, 2011)(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011).

### 3.4.1 Resposta das Questões de Pesquisa

Nesta seção, as respostas para as questões de pesquisa definidas na Seção 3.2 são apresentadas e discutidas.

a) *QP1: Quais características de qualidade são utilizadas para avaliações de IHC em sistemas ubíquos?*

Para essa questão foram extraídas as características presentes nos modelos e/ou *frameworks* citados anteriormente, as características listadas nos artigos que não estavam organizadas em um modo hierárquico e as características apresentadas como importantes na avaliação da interação entre um usuário e um sistema ubíquo, mas que não foram formalmente definidas como características de qualidade. Todas essas informações foram extraídas como características de qualidade na análise para responder a essa pergunta. Algumas delas estão apresentadas na Tabela 3.6 e a síntese de todas será apresentada na Seção 3.5.

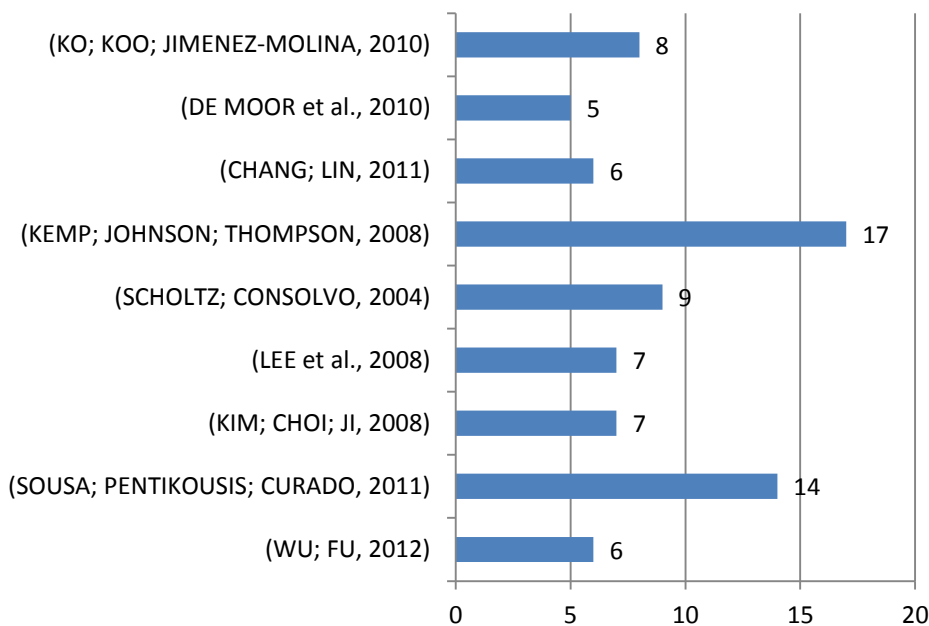
**Tabela 3.6.** Exemplo de Características Extraídas

Referência	Características
(ABI-CHAR et al., 2010)	Confiança, Segurança, Privacidade
(ZHANG et al., 2006)	Usabilidade, Status da Rede, Capacidade do Dispositivo
(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)	Ubiquidade, Sensibilidade ao Contexto, Difusão
(CAPPIELLO; PUGLIA; VITALETTI, 2009)	Usabilidade
(RANGANATHAN et al., 2005)	Sensibilidade ao Contexto, Segurança, Usabilidade
(KIM; CHOI; JI, 2008)	Adaptabilidade, Controlabilidade, Interconectividade, Mobilidade, Previsibilidade, Simplicidade, Transparência

Foram encontradas 134 características, a Figura 3.7 apresenta a distribuição de características extraídas pelos artigos que mais apresentaram características de qualidade.

Os artigos encontrados possuem algumas limitações com relação às características extraídas. Alguns propõem somente uma lista de características, mas sem uma clara definição delas (KIM; CHOI; JI, 2008) (SOUSA; PENTIKOUSIS; CURADO, 2011) (ZHANG et al., 2006). Quando definidas, não existe um consenso sobre as características e vários trabalhos utilizam diferentes nomes para o mesmo conceito, por exemplo, (KIM; CHOI; JI, 2008)

apresenta a característica *Transparência*, entretanto, (KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008) a define como *Difusão*.



**Figura 3.7** - Número de Características Extraídas por Estudo

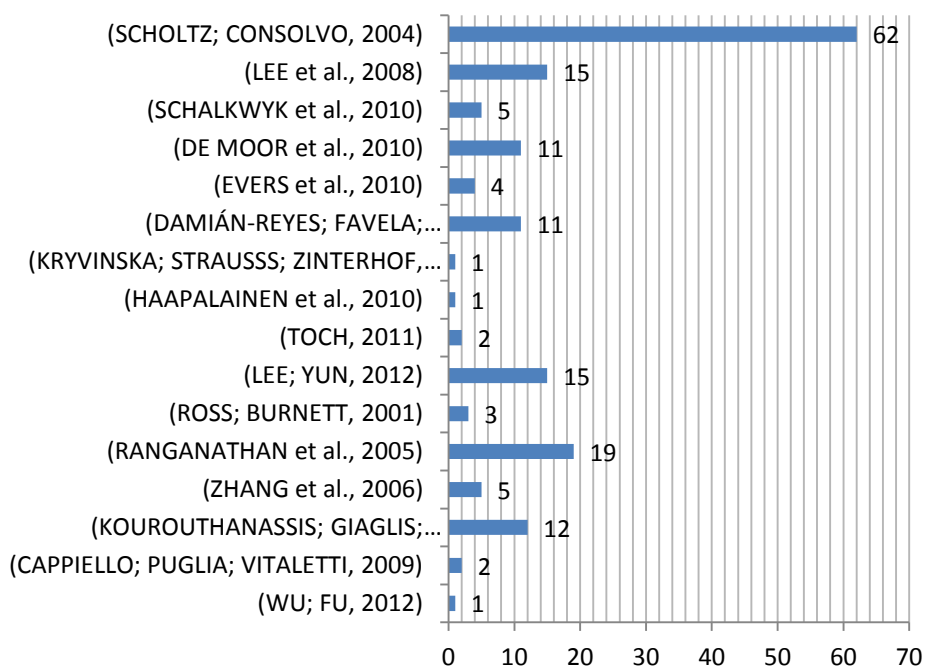
Outro exemplo é sobre *Sensibilidade ao Contexto*, alguns artigos definem como *Suporte à Contextualização* (LEE; YUN, 2012), *Adaptabilidade* (KIM; CHOI; JI, 2008) ou mesmo *Invisibilidade* (SCHOLTZ; CONSOLVO, 2004). Assim, foi necessário realizar uma análise de sinônimos das características, verificando as definições de cada conceito com o intuito de agrupar em somente um nome de característica e também retirar características duplicadas e que não possuem uma definição coerente com a literatura clássica. Essa análise é apresentada na Seção 3.5.

Outra limitação é que alguns artigos não possuem novas características de IHC para avaliar os sistemas ubíquos, somente características tradicionais, como *Usabilidade* (ABI-CHAR et al., 2010) (ROSS; BURNETT, 2001)(CAPPIELLO; PUGLIA; VITALETTI, 2009). Outros artigos que propõem novas características (sensibilidade ao contexto, transparência), não as aplicam em avaliações de qualidade (KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008) (KIM; CHOI; JI, 2008). Todas as características que foram consideradas pertinentes para a IHC foram analisadas e sumarizadas em uma lista final que é apresentada na Seção 3.5.

b) *QP2: Quais são as medidas de software utilizadas para a avaliação de IHC nesses sistemas?*

Dos 32 estudos, somente 16 apresentaram medidas de *software* para avaliar sistemas ubíquos, porém a maioria desses não apresenta a descrição completa ou qualquer detalhe de

como calculá-la. A Figura 3.8 apresenta o número de medidas de *software* presentes em cada estudo. No total, foram obtidas 155 medidas, que são apresentadas no Apêndice A.



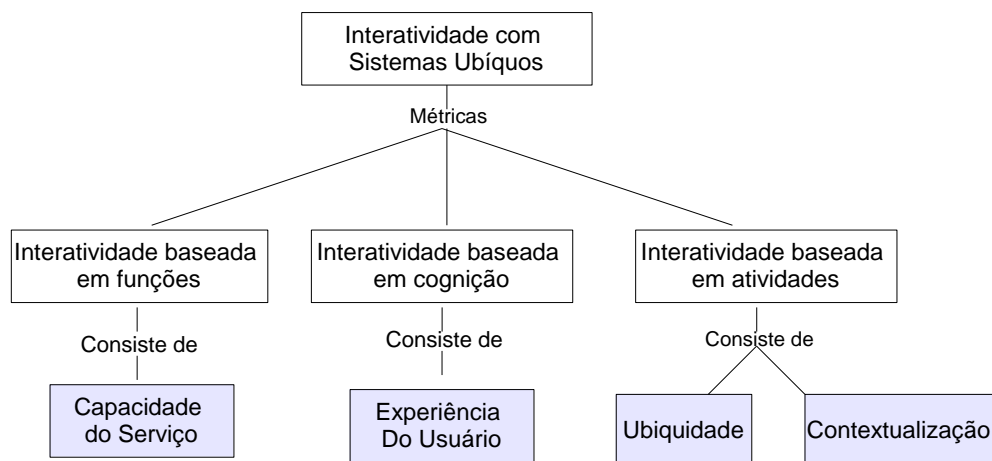
**Figura 3.8** – Número de Medidas de *Software* Extraídas por Estudo

Vale ressaltar, que neste gráfico, a soma total do número de medidas é maior que 155 medidas, pois foi identificado que alguns artigos possuem as mesmas medidas de *software*.

c) *QP3: Existem modelos de qualidade para avaliação da IHC em Sistemas Ubíquos?*

Para responder essa questão, foi analisado se o artigo possuía medidas, características e subcaracterísticas organizadas de modo hierárquico, assim como a norma SQuaRE recomenda (ISO/IEC 25000, 2005). Neste caso, pode ser um *framework* ou modelo empírico. Somente dois entre os artigos encontrados descrevem uma solução aproximada com a norma: (LEE et al., 2008) e (SCHOLTZ; CONSOLVO, 2004). Apesar desses trabalhos não chamarem explicitamente tais soluções de modelos de qualidade, eles as utilizam para fazer avaliações, assim foi considerado como modelos de qualidade.

O estudo desenvolvido por (LEE et al., 2008) visa o desenvolvimento de medidas centradas no usuário para avaliação de atributos de interatividade em sistemas ubíquos. Os atributos de avaliação definidos em seu modelo são apresentados na Figura 3.9.



**Figura 3.9** - Estrutura de Avaliação da Interatividade, traduzida de (LEE et al., 2008)

De acordo com (LEE et al., 2008), *Capacidade do Serviço* diz respeito ao nível de desempenho, segurança (proteção ao usuário e prevenção de erros), velocidade e armazenamento. *Experiência do Usuário* representa a participação e o grau de esforço para utilizar o sistema ubíquo. *Ubiquidade* representa a disponibilidade do sistema ubíquo e, por fim, *Contextualização* é o nível de provisão de serviços customizados através da percepção de status.

O modelo proposto por (LEE et al., 2008) possui algumas limitações. Em primeiro lugar, ele não tem uma boa hierarquia: alguns atributos possuem muita informação (por exemplo, capacidade de serviço), este deveria ser decomposto em subfatores, para facilitar sua avaliação. Segundo, ele falha em definir características essenciais para fazer avaliações de IHC de sistemas ubíquos. Por exemplo, ele não menciona nenhuma informação sobre interação calma (*calmness*) (RIEKKI; ISOMURSU; ISOMURSU, 2004), que é uma característica muito importante, pois tem impacto direto na interação do usuário. Se o usuário receber constantemente informações e serviços não relevantes, dificilmente ele vai usá-lo para suas atividades diárias. Em suma, ele não apresenta medidas para avaliar todos os atributos propostos em seu modelo.

O segundo estudo (SCHOLTZ; CONSOLVO, 2004) visa desenvolver um framework focado no usuário para avaliação de aplicações ubíquas. Ele define nove áreas de avaliação, que foram divididas em métricas, que por sua vez foram divididas em medidas conceituais. Uma dessas áreas propostas refere-se à característica *Atenção*, apresentada na Tabela 3.7.

**Tabela 3.7 - Framework de Avaliação de Sistemas Ubíquos, traduzida de (SCHOLTZ; CONSOLVO, 2004)**

Área	Métricas	Medidas Conceituais
Atenção	Foco	Número de vezes que o usuário deve mudar o foco para a tecnologia, número de eventos não observados pelo usuário.
	Sobrecarga	Percentual de tempo que o usuário gasta mudando de foco, carga de trabalho imposta no usuário para foco.

Essa área se refere à maior consciência a um evento ou ação específica para selecioná-lo para o aumento do processamento. Em sistemas ubíquos, essa característica se refere à consciência na tarefa principal e não na tecnologia. O trabalho divide atenção em duas métricas: foco e sobrecarga. Foco se refere ao ponto de atenção do usuário, os *designers* devem investigar o que desvia a atenção do usuário da sua atividade principal. Sobrecarga se refere ao tempo perdido introduzido pela tecnologia.

A maior limitação deste trabalho é a definição deficiente das medidas, por exemplo, elas possuem apenas nomes. Propriedades como função de medição, elementos de medição e interpretação não são definidos nas medidas.

Concluindo, através das respostas a essa questão, percebe-se uma ausência de trabalhos na literatura que tenham definido um modelo de qualidade que possua características essenciais para IHC em sistemas ubíquos e medidas completas para realizar medições de software.

### 3.4.2 O Mapa Sistemático

Além de responder às questões de pesquisa, a fase Resultados envolve também a construção de um mapa sistemático, que como mencionado na Seção 3.1, é um gráfico de bolhas com pelo menos dois eixos, onde o tamanho de cada bolha é proporcional ao número de estudos presentes na intersecção dos dois eixos. Esses eixos são as facetas já definidas pelo *keywording* de tópicos relevantes, apresentado na Seção 3.3.3. A primeira faceta corresponde ao tipo de contribuição (apresentado na Tabela 3.3), a segunda é o tipo de pesquisa (apresentado na Tabela 3.4). As Figuras 3.10 e 3.11 apresentam o resultado dessa classificação e o Apêndice B apresenta a classificação de todos os estudos nas facetas.

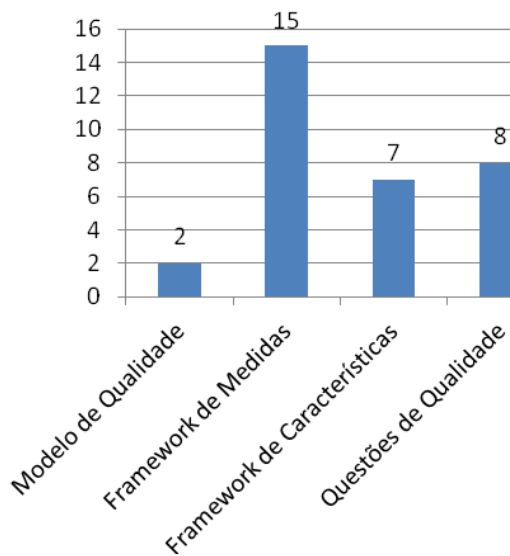


Figura 3.10 - Distribuição por tipo de contribuição

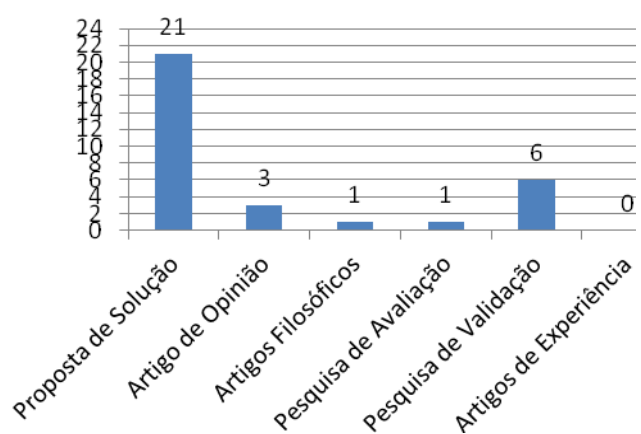


Figura 3.11 - Distribuição por tipo de pesquisa

Por fim, esses dados foram cruzados e o mapa sistemático foi criado. Segundo (PETERSEN et al., 2008), essa análise permite apresentar as frequências dos estudos em cada categoria.

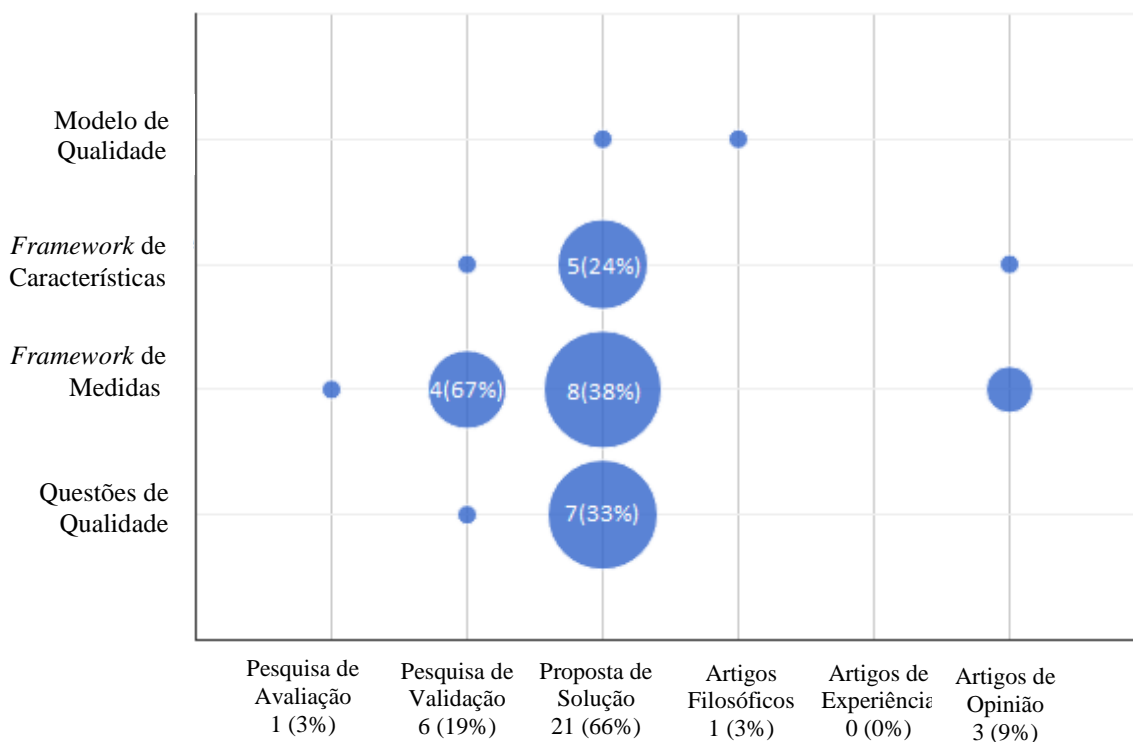


Figura 3.12 - O Mapa Sistemático

Analisando a Figura 3.12, foi concluído que:

- A maioria dos artigos (66%) são “*Propostas de Solução*”. Nenhum estudo foi utilizado na prática (*Artigos de Experiência*). Somente seis (19%) se classificam como “*Pesquisa de Validação*”, isto é, foram utilizados experimentos para validação do trabalho e somente um (3%) é “*Pesquisa de Avaliação*”, o que significa que a proposição é amplamente utilizada no mercado;
- Dos artigos que são “*Proposta de Solução*”, a maioria se classifica como “*Framework de Medidas*” (8 - 38%), “*Questões de Qualidade*” (7 - 33%) e “*Framework de Característica*”(5 - 24%). No entanto, esses trabalhos que contém medidas não detalham completamente como medir o sistema ubíquo, não apresentam as funções de medição, valores de interpretação e métodos de coleta, não sendo possível, portanto, fazer avaliações da qualidade. Já os trabalhos que propõem características ou questões de qualidade não apresentam um padrão em suas definições, alguns não fornecem uma clara definição e outros não levam em consideração importantes características específicas dos sistemas ubíquos;
- Foram encontrados somente dois “*Modelos de Qualidade*”, que é considerada a contribuição mais completa por possuir características, subcaracterísticas e medidas; e
- Dos modelos de qualidade encontrados, um se classifica como *Proposta de Solução* e o outro como o único “*Artigos Filosóficos*”, portanto, não existe um modelo de qualidade para sistemas ubíquos que tenha sido validado e/ou utilizado na prática.

Através dessa análise é possível afirmar que a área de medição da IHC em sistemas ubíquos ainda é muito recente. Mais pesquisas são necessárias para desenvolver soluções que agreguem as características existentes e definam medidas de *softwares* que estejam aptas a avaliar essas características em um sistema ubíquo.

A Seção 3.5 a seguir apresenta uma tentativa de agregar as características existentes encontradas por esse MS em um conjunto final de características de qualidade para avaliar a IHC em sistemas ubíquos.



### 3.5 Definição do Conjunto Final de Características de Qualidade

Como apresentado na Seção 3.4.1, foram encontradas 134 características. No entanto, existem características duplicadas com o mesmo nome e significado e também características com o mesmo significado, mas com nomes diferentes. Para obter o conjunto final de características, foi realizada uma análise dos significados (análise semântica) considerando o que estava escrito nos artigos encontrados pelo MS e também na literatura clássica das áreas envolvidas.

Essa análise foi executada em três passos. No primeiro passo somente um pesquisador identificou os sinônimos das características e depois essa proposta inicial foi revisada por pares (dois outros pesquisadores executaram a revisão dessa proposta). Nessa revisão em pares (*peer review*), os pesquisadores identificavam se concordavam com a identificação dos sinônimos da característica ou não, caso não, uma nova proposta era feita. Após as duas revisões, uma reunião foi realizada para chegar a um consenso. No final, 26 características pertinentes para avaliação de IHC em sistemas ubíquos foram identificadas, tais características são apresentadas na Tabela 3.8.

**Tabela 3.8** – Lista de Características de Qualidade da IHC em Sistemas Ubíquos

Característica	Definição	Sinônimos	Referência
Aceitabilidade	Representa a vontade para usar um aplicativo e as taxas de utilização.	<i>Adoption</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Potential customer acceptance</i>	(ROSS; BURNETT, 2001)
		<i>Acceptability</i>	(WAIBEL et al., 2010)
Atenção	Habilidade de manter a atenção do usuário para sua atividade principal e não no sistema e na tecnologia envolvida.	<i>Attention</i>	(WU; FU, 2012)(SCHOLTZ; CONSOLVO, 2004)
		<i>Distraction</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Focus</i>	(KEMP; THOMPSON; JOHNSON, 2008) (HAAPALAINEN et al., 2010)

**Tabela 3.8** - Lista de Características de Qualidade da IHC em Sistemas Ubíquos (Continuação)

Capacidade do Dispositivo	Propriedades do dispositivo onde a aplicação será executada: tamanho da tela, intensidade da cor, vida da bateria.	<i>Device Capability</i>	(ZHANG et al., 2006)
		<i>Device</i>	(DE MOOR et al., 2010)
		<i>Required HW/SW</i>	(WU; FU, 2012)
Capacidade da Rede	Representa a coleta de informações de infraestrutura de rede: força do sinal, delay, jitter.	<i>Network Status</i>	(ZHANG et al., 2006)
		<i>Infrastructure</i>	(DE MOOR et al., 2010)
		<i>Network</i>	(DE MOOR et al., 2010)
Calma	Habilidade do sistema de prevenir os usuários de se sentirem sobrecarregados por informações.	<i>Calm technology</i>	(WAGNER; TOFTEGAARD; BERTELSEN, 2012)
		<i>Aesthetic and minimalist design</i>	(KEMP; THOMPSON; JOHNSON, 2008)
Confiabilidade	A capacidade do software de manter um nível específico de desempenho quando utilizado sob condições específicas.	<i>Reliability</i>	(WAIBEL et al., 2010)
Confiança	É a crença do usuário que o sistema utilizará seus dados apropriadamente e não causará nenhum dano. Implica questões de consciência, privacidade e controle.	<i>Trust</i>	(SOUSA; PENTIKOUSIS; CURADO, 2011)(ABI-CHAR et al., 2010)(JIA; COLLINS; NIXON, 2009)(SUN; DENKO, 2008)(SCHOLTZ; CONSOLVO, 2004)(EVERS et al., 2010)
		<i>Trust/Ethics/Responsibility</i>	(KEMP; THOMPSON; JOHNSON, 2008)
Disponibilidade	O serviço está sempre disponível, independentemente de hardware, software ou falhas do usuário, é muitas vezes tida como certa até que o tempo de inatividade ocorre.	<i>Availability</i>	(KRYVINSKA; STRAUSSS; ZINTERHOF, 2011)(KO; KOO; JIMENEZ-MOLINA, 2010)
Eficácia	Refere-se à completude na realização de tarefas.	<i>Effectiveness</i>	(SCHOLTZ; CONSOLVO, 2004)(KEMP; THOMPSON; JOHNSON, 2008)

**Tabela 3.8** - Lista de Características de Qualidade da IHC em Sistemas Ubíquos (Continuação 2)

Eficiência	Refere-se à quantidade de esforço e recursos necessários para se chegar a um determinado objetivo no sistema.	<i>Performance</i>	(CAPPIELLO; PUGLIA; VITALETTI, 2009)(SCHALKWYK et al., 2010)
		<i>Efficiency</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Timeliness</i>	(KEMP; THOMPSON; JOHNSON, 2008)
		<i>User's performance</i>	(RANGANATHAN et al., 2005)
Escalabilidade	Habilidade de prover serviços para poucos ou um grande número de usuários.	<i>Scalability</i>	(SCHOLTZ; CONSOLVO, 2004)
Facilidade de Uso	O sistema deve ser fácil para o uso por um grupo alvo de usuários.	<i>Ease of Use</i>	(CAPPIELLO; PUGLIA; VITALETTI, 2009)(KEMP; THOMPSON; JOHNSON, 2008)
		<i>Facility of Use</i>	(RUBIO; BOZO, 2007)
		<i>Easy to Use</i>	(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)
		<i>Perceived Ease of Use</i>	(CHANG; LIN, 2011)
Familiaridade	As interações do usuário com o sistema devem melhorar a qualidade do seu trabalho. O usuário deve ser tratado com respeito. O projeto deve ser esteticamente prazeroso.	<i>Familiarity</i>	(IQBAL et al., 2005)
Interconectividade	Rede interconectada entre dispositivos permite compartilhar informações.	<i>Interconnectivity</i>	(KIM; CHOI; JI, 2008)

**Tabela 3.8** - Lista de Características de Qualidade da IHC em Sistemas Ubíquos (Continuação 3)

Mobilidade	Capacidade do sistema de prover aos usuários acesso contínuo aos recursos e informações, independente de sua localização dentro do limite dos sistemas.	<i>Mobility</i>	(KIM; CHOI; JI, 2008)
		<i>Ubiquity</i>	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
		<i>Flexibility</i>	(KEMP; THOMPSON; JOHNSON, 2008)
		<i>Multi-space support</i>	(KO; KOO; JIMENEZ-MOLINA, 2010)
Previsibilidade	Habilidade de, a partir de experiências passadas, prever o resultado da execução do sistema.	<i>Predictability</i>	(KIM; CHOI; JI, 2008)
Privacidade	Habilidade de manter informações e dados protegidos.	<i>Privacy</i>	(WU; FU, 2012)(ABI-CHAR et al., 2010)(SUN; DENKO, 2008)(SCHOLTZ; CONSOLVO, 2004)(JAFARI et al., 2010)(TOCH, 2011)(LIAMPOTIS et al., 2009)
Proteção (Safety)	O nível de risco de danos a pessoas, negócios, <i>software</i> , <i>hardware</i> , bens ou o ambiente em um contexto de uso especificado.	<i>Driver safety</i>	(ROSS; BURNETT, 2001)
Robustez	Grau com que um sistema ou componente pode funcionar corretamente na presença de entradas inválidas ou condições ambientais estressantes.	<i>Application Robustness</i>	(SCHOLTZ; CONSOLVO, 2004)
Satisfação do usuário	O grau de aceitação do usuário e quão atrativo o sistema é para o usuário	<i>User Satisfaction</i>	(SOUSA; PENTIKOUSIS; CURADO, 2011)
		<i>Customer Satisfaction</i>	(CAPPIELLO; PUGLIA; VITALETTI, 2009)
		<i>Appeal</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>User Satisfaction</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Satisfaction</i>	(RANGANATHAN et al., 2005)

**Tabela 3.8** - Lista de Características de Qualidade da IHC em Sistemas Ubíquos (Continuação 4)

Segurança	A proteção do transporte e armazenamento de informações e controles de segurança de quem pode acessar, utilizar e/ou modificar informações de contexto.	<i>Security</i>	(WU; FU, 2012)(ABI-CHAR et al., 2010)(SUN; DENKO, 2008)(RANGANATHAN et al., 2005)
Sensibilidade ao Contexto	Capacidade do sistema de perceber informações contextuais proativamente e adaptar suas funcionalidades.	<i>Context-Awareness</i>	(LEE et al., 2008)(LEE; YUN, 2012)(TOCH, 2011)(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)(KO; KOO; JIMENEZ-MOLINA, 2010)
		<i>Contextualization</i>	(LEE et al., 2008)(LEE; YUN, 2012)
		<i>Context Sensitivity</i>	(RANGANATHAN et al., 2005)
		<i>Adaptivity</i>	(KO; KOO; JIMENEZ-MOLINA, 2010)
Simplicidade	A interface do usuário e as instruções são simples	<i>Simplicity</i>	(KIM; CHOI; JI, 2008)
Transparência	Habilidade do sistema de esconder componentes computacionais e, assim, os usuários podem não estar cientes dos mesmos. Além disso, a interação é executada através de interfaces naturais.	<i>Diffusion</i>	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
		<i>Invisibility</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Interaction Transparency</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Invisibility</i>	(KEMP; THOMPSON; JOHNSON, 2008)
		<i>Understandability</i>	(THOMPSON; AZVINE, 2004)
<i>Transparency</i>	(KO; KOO; JIMENEZ-MOLINA, 2010)		
Usabilidade	A capacidade do software de ser entendido, aprendido, utilizado e atrativo para o usuário, quando utilizado sob condições específicas	<i>Usability</i>	(IQBAL et al., 2005)(ROSS; BURNETT, 2001)

**Tabela 3.8** - Lista de Características de Qualidade da IHC em Sistemas Ubíquos (Continuação 5)

Utilidade	Habilidade de oferecer utilidade para o usuário. Fornecer uma contribuição para o usuário que antes do sistema não estava disponível.	<i>Utility</i>	(SOUSA; PENTIKOUSIS; CURADO, 2011)(CHANG; LIN, 2011)
		<i>Usefulness</i>	(IQBAL et al., 2005)
		<i>Impact and Side Effects</i>	(SCHOLTZ; CONSOLVO, 2004)
		<i>Perceived Usefulness</i>	(WAIBEL et al., 2010)

É possível notar que existem características que são claramente subcaracterísticas de outra, como por exemplo, as características *Facilidade de Uso, Eficiência, Eficácia, Satisfação, Familiaridade e Simplicidade* são claramente componentes da característica *Usabilidade; Segurança e Privacidade* podem impactar na característica *Confiança*.

Também é importante salientar, a diferença de *Utilidade* para *Usabilidade*. Utilidade, segundo Nielsen (NIELSEN, 1994), diz respeito ao sistema capaz de fazer o que é necessário para o usuário e usabilidade diz respeito ao quão bem os usuários podem desempenhar as funcionalidades de um sistema. Todos eles impactam na *Aceitabilidade* do sistema pelo usuário.

Muitas dessas características são apresentadas em normas internacionais como SQuaRE (ISO/IEC 25000, 2005), ISO 9126 (ISO/IEC 9126, 2001) e ISO 9241 (ISO 9241-11, 1998). Por exemplo: Usabilidade, Confiabilidade, Proteção, Desempenho, Segurança, Disponibilidade e, claramente, percebe-se que elas podem ser utilizadas para avaliar qualquer tipo de produto. Por outro lado, algumas características não são definidas nessas normas e, provavelmente, são particulares para a interação com esse tipo de sistema (ubíquo), como por exemplo: sensibilidade ao contexto, foco e mobilidade.

### 3.6 Considerações Finais

Neste capítulo foi apresentado um estudo de MS da literatura. Este estudo foi realizado para investigar e definir um conjunto de características de qualidade da IHC específicas para os sistemas ubíquos. Após a extração de dados, uma análise do significado das características encontradas foi realizada com o intuito de agrupar conceitos similares e retirar características que estavam duplicadas e/ou definidas de forma errada. Ao final dessa análise, um conjunto de 26 características foi definido.

Além da definição desse conjunto de características, com a realização desse mapeamento foi possível observar que não existem Modelos de Qualidade específicos para avaliar os sistemas ubíquos, o que existem são *frameworks* com características, subcaracterísticas e medidas que ainda estão incompletas e necessitam de uma melhor definição, para ser possível fazer uma avaliação da qualidade.

Por fim, o mapeamento retornou um conjunto de medidas de *software* que, como citado anteriormente, não estão completamente definidas, pois não possuem qualquer fórmula de cálculo e apoio na sua interpretação. Tais medidas estão listadas no Apêndice A.

# 4 Medidas de *Software* para Avaliação da IHC em Sistemas Ubíquos

Este capítulo apresenta a definição das medidas de *software* propostas para avaliação de um conjunto de características selecionadas das vinte e seis características da IHC em sistemas ubíquos definidas no capítulo anterior. Para a definição dessas medidas, a abordagem GQM é utilizada (apresentada na Seção 2.3.5). Como apresentado, o GQM consiste de quatro fases bem definidas: Planejamento (Seção 4.2), Definição do objetivo de medição, das questões e medidas (Seção 4.3), Coleta de Dados (Seção 4.4) e Interpretação (Seção 4.5). Por fim, a Seção 4.6 apresenta as considerações finais desse capítulo.

## 4.1 Seleção de Características Específicas de IHC para Sistemas Ubíquos

O foco deste trabalho é definir medidas de *software* para avaliar características essenciais e específicas de IHC em sistemas ubíquos. A lista de 26 características definida na Seção 3.5 possui características de IHC que qualquer sistema deve apresentar (*e.g.*, usabilidade e segurança), entretanto, está fora do escopo desse trabalho definir medidas de *software* para avaliação de características de qualidade tradicionais da IHC. Sendo assim, foi necessário selecionar características que fossem mais específicas para sistemas ubíquos. Essa seleção foi baseada nas definições apresentadas na Seção 2.1, as quais são rerepresentadas a seguir com palavras em negrito que se referem à definição do (WEISER, 1991):

- “Computação ubíqua representa o conceito de computação em todo o lugar, fazendo com que o uso da computação e a comunicação sejam **transparentes** para o usuário.” de (YAU; WANG; KARIM, 2002);
- “Computação ubíqua é o uso de um conjunto de computadores dos mais variados tamanhos, formatos e funções, que de forma coordenada e autônoma, auxiliam as pessoas na realização das diversas **tarefas cotidianas**. Esse auxílio é realizado de tal forma que a **infraestrutura computacional responsável fica escondida** no ambiente.” de (LIMA, 2011);
- “A computação ubíqua está presente quando **serviços computacionais tornam-se**



**disponíveis** para os usuários de tal forma que o computador **não é mais uma ferramenta visível ou essencial** para acessar esses serviços. Assim, os serviços computacionais são **acessados em qualquer tempo ou lugar**, de forma **transparente**, através do uso de **dispositivos comuns.**” de (SPÍNOLA; MASSOLLAR; TRAVASSOS, 2007).

- “A computação ubíqua é esperada para ter um enorme efeito nas vidas das pessoas. Computadores estarão em **todos os lugares ao redor dos usuários**, conectados uns aos outros e servindo aos usuários. Entretanto, para se tornar um sucesso, a interação com a aplicação ubíqua deve ser **calma.**” de (RIEKKI; ISOMURSU; ISOMURSU, 2004)

As palavras em negrito refletem algumas características definidas anteriormente, o fato do sistema auxiliar em tarefas cotidianas, ser transparente, calmo e com isso manter o foco do usuário nas suas atividades principais e ser acessado em qualquer lugar ou tempo refletem as características: *Sensibilidade ao Contexto, Transparência, Atenção, Calma e Mobilidade.*

Além disso, foi observado em trabalhos do GREat (LIMA, 2011; LIMA et al., 2011) que *Sensibilidade ao Contexto, Mobilidade e Transparência* são requisitos essenciais para o desenvolvimento dos sistemas ubíquos. Assim, o subconjunto de cinco características (*Sensibilidade ao Contexto, Transparência, Atenção, Calma e Mobilidade*) foi selecionado para a definição das medidas de *software*. Essas cinco características selecionadas estão descritas abaixo.

*Sensibilidade ao Contexto*, como já amplamente apresentado na Seção 2.1.3, corresponde à capacidade de coletar informações contextuais e utilizar essas informações para realizar as adaptações (KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008). É importante deixar claro que existem outros tipos de adaptações, como as adaptações estáticas que modificam a estrutura do *software* devido a mudanças em requisitos de *software*. Entretanto, nesse trabalho o conceito de sensibilidade ao contexto está atrelado com as adaptações dinâmicas, aquelas em que o *software* modifica seu comportamento devido a variações de contexto (ROCHA, 2007).

*Transparência*, segundo (SATYANARAYANAN, 2001), pode ser alcançada pela proatividade do sistema, de forma que o usuário seja minimamente distraído. Isso acontecerá se ele conhecer bem o usuário, suas expectativas e seu ambiente. Além disso, o sistema ubíquo deve esconder sua infraestrutura computacional no ambiente do usuário, de forma que o usuário não perceba que está interagindo com um conjunto de computadores.

*Atenção* não é uma característica nova, ela foi bastante estudada na área da computação *desktop* e representa uma maior atenção para a atividade no computador (SCHOLTZ;

CONSOLVO, 2004). No entanto, na computação ubíqua, essa característica significa o contrário. No ambiente ubíquo, os computadores estão escondidos e substituem as atividades do usuário, possibilitando assim o usuário focar nas diversas atividades mentais e físicas como andar, dirigir ou outras interações do mundo real (GARLAN; SIEWIOREK; STEENKISTE, 2002). Assim, na computação ubíqua, *Atenção* representa um aumento do foco de atenção do usuário nas suas atividades do dia-a-dia.

*Calma* é visionada por Mark Weiser como uma nova abordagem para encaixar apropriadamente a computação nas vidas das pessoas. No artigo de (RIEKKI; ISOMURSU; ISOMURSU, 2004), o autor afirma que calma significa “livre de distração, excitação ou perturbação” e que uma aplicação calma é aquela que está disponível a qualquer hora e lugar, interage com o usuário no momento correto, somente apresenta informações relevantes, utiliza a periferia da atenção e o centro, somente quando necessário e é fácil e natural de utilizar e se encaixa bem na situação do usuário.

*Mobilidade* não é uma característica nova, ela foi considerada desde os anos 60, através de sistemas de compartilhamento de tempo, passando para a computação distribuída nos anos 70, até a computação móvel, nos anos 90. Na computação móvel, por causa do desenvolvimento de dispositivos móveis (*e.g. laptops*, PDAs e celulares) e de tecnologias de comunicação sem fio (*e.g. 3G, 4G e wifi*), a computação passou a fazer parte do cotidiano das pessoas, tornando o acesso móvel uma realidade. Já na era da computação ubíqua, caracterizada pela mínima distração do usuário, existe uma busca pela mobilidade “*seamless*” que se refere ao uso contínuo ou ininterrupto da computação enquanto o usuário se move através de dispositivos (YU et al., 2013).

## 4.2 Planejamento para Elaboração das Medidas

Os principais objetivos da fase de planejamento são: coletar todas as informações necessárias para uma avaliação de sucesso e preparar e motivar a equipe para a avaliação. Faz-se necessário, então, estabelecer um time GQM, uma área de melhoria, um projeto de aplicação e o time do projeto. Além de realizar o treinamento com todos os participantes.

Para este trabalho, o time GQM consistiu na equipe de pesquisa desse trabalho, nos participantes do projeto *Maximum*<sup>4</sup> e outros especialistas em sistemas ubíquos. No total, nove pessoas participaram. As reuniões foram feitas em grupos ou individualmente, de acordo com a disponibilidade dos participantes. No total, foram realizadas cinco reuniões GQM.

---

<sup>4</sup> <http://www.great.ufc.br/maximum/>

A área de melhoria foi definida a partir da motivação e do objetivo deste trabalho: avaliação da qualidade da IHC em sistemas ubíquos. Para o projeto de aplicação não foi considerado um projeto ou sistema específico e sim características dos sistemas ubíquos em geral (apresentadas na Seção 2.1.2), portanto, não foi definido o time do projeto.

Um treinamento sobre medições de *software* foi realizado com todas as pessoas que participaram das reuniões GQM, a fim de deixar toda a equipe motivada e informada sobre os princípios da medição de *software*. Após essas decisões, a fase de definição de medidas foi iniciada, a qual é apresentada na próxima seção.

## 4.3 Definição das Medidas

Nesta fase do GQM, os objetivos de medição, as questões e as medidas de *software* são definidos e documentados. As próximas subseções apresentam cada um desses pontos.

### 4.3.1 Objetivo de Medição

Para a definição do objetivo de medição foi utilizado o *template* definido por (BASILI; ROMBACH, 1994), apresentado na Seção 2.3.5. De acordo com esse *template*, devem ser definidos os objetos sob medição, os propósitos do objetivo, os focos de qualidade do objeto que a medição foca e o ponto de vista (quem mede o objeto).

A especificação do objetivo de medição desse trabalho está apresentada na Tabela 4.1. O objeto sob medição é o sistema ubíquo, o propósito é avaliar e os focos de qualidade são as cinco características de qualidade da IHC para sistemas ubíquos, selecionadas anteriormente: Sensibilidade ao Contexto, Mobilidade, Transparência, Atenção e Calma. Por fim, o ponto de vista é do desenvolvedor e do usuário.

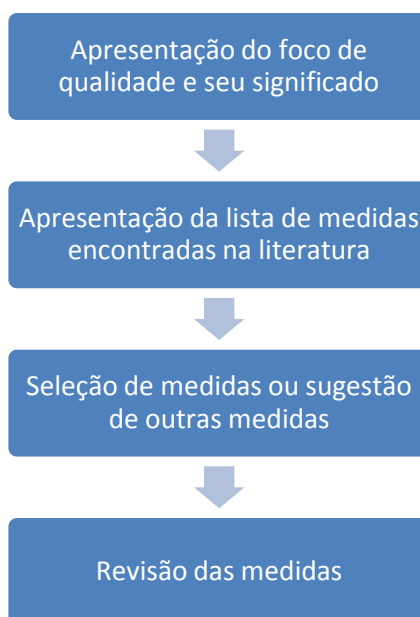
Tabela 4.1 – Objetivo de Medição	
<b>Analisar</b>	o sistema ubíquo
<b>Com o propósito de</b>	avaliar
<b>Com respeito à</b>	sensibilidade ao contexto, mobilidade, transparência, atenção e calma
<b>Ponto de vista</b>	desenvolvedor e usuário

Para a definição das questões e das medidas desse objetivo de medição, entrevistas GQM foram realizadas a fim de se obter informações importantes. Para isso foi utilizado o *abstract sheet*

(apresentado na Figura 2.14), um tipo de *template* que é estruturado para extrair informações dos especialistas do objeto sob medição.

### 4.3.2 Entrevistas GQM

As entrevistas GQM foram realizadas em grupo e individualmente, pois dependia da disponibilidade dos entrevistados. Para cada pessoa (caso fosse individualmente) ou grupo, as atividades descritas na Figura 4.1 foram realizadas.



**Figura 4.1** – Fluxo de Atividades das Entrevistas GQM

A primeira atividade consistiu na apresentação do foco de qualidade e seu significado. Cada foco de qualidade foi apresentado separadamente e para cada um foi utilizado um *abstract sheet* (Figura 2.14) para documentar suas medidas. Assim foram gerados cinco *abstracts sheets*.

A segunda atividade consistiu na apresentação de uma lista de medidas encontradas na literatura através do MS realizado. Essa lista de medidas está apresentada no Apêndice A.

A terceira atividade consistiu na seleção de medidas dessa lista. Nessa seleção, os especialistas ficaram livres para modificar a medida ou completá-la, dado que muitas medidas possuíam apenas o nome. Cada pessoa ou grupo seguiu esse mesmo fluxo de atividades, assim, todos sempre recebiam os *abstracts sheets* em branco.

A quarta e última atividade consistiu na revisão das medidas por um especialista escolhido por ser o mais experiente na área da computação ubíqua.

Os *abstracts sheets* gerados nessas entrevistas são apresentados na Seção 4.3.3 a seguir.

### 4.3.3 Abstract Sheets

Cada foco da qualidade no *abstract sheet* é representado por um número. Para cada um dos focos podem existir hipóteses de *baseline* e fatores de variação e cada um é representado pelo mesmo número do foco de qualidade correspondente. Cada fator de variação possui um impacto na hipótese de *baseline*. Essa correspondência também é feita pela mesma numeração. Um exemplo de leitura do *abstract sheet*: para o foco da qualidade “**1**) Corretude da Adaptação” existe o seguinte fator de variação: “**1.1**) Variedade de Informações Contextuais”, entre outros. O fator de variação “**1.1**” possui o impacto na hipótese de *baseline* também enumerado por “**1.1**”. O primeiro *abstract sheet* é para “sensibilidade ao contexto” e está apresentado abaixo.

Objeto	Propósito	Foco da Qualidade	Ponto de vista
Sistema Ubíquo	Avaliar	Sensibilidade ao contexto	Especialistas e Usuário
<p><b>Foco da qualidade</b></p> <p>1) Corretude da Adaptação</p> <p>2) Grau de concordância com a adaptação: n° de vezes que vários usuários concordam com uma adaptação</p> <p>3) Granularidade do Contexto: informação crua, situações de contexto, informações de alto nível</p> <p>4) Tempo de Adaptação (tempo que demora para se adaptar)</p> <p>5) Grau de interferência do usuário para a adaptação ocorrer: X/Y, onde X=Quantidade de adaptações que o usuário precisa intervir Y=Quantidade total de adaptações</p>		<p><b>Fatores de variação</b></p> <p>1.1) Variedade de informações contextuais</p> <p>1.2) Dependência de recursos externos</p> <p>1.3) Corretude da informação e situações de contexto capturada</p> <p>1.4) Frequência de mudanças do contexto</p> <p>4.1) Conexão de rede</p> <p>5.1) Uso de técnicas para raciocinar sobre as incertezas: lógica <i>fuzzy</i>, lógica probabilística, rede bayesiana, modelo oculto de Markov, teoria de Dempster-Shafer.</p>	
<p><b>Hipóteses de <i>Baseline</i></b></p> <p>1) Quanto maior, melhor</p> <p>2) Quanto maior, melhor</p> <p>3) Quanto maior é o nível de informação de contexto que a aplicação consegue processar, melhor será para a aplicação</p> <p>4) Quanto menor, melhor</p> <p>5) Quanto maior o valor, pior</p>		<p><b>Impacto na Hipótese de <i>Baseline</i></b></p> <p>1.1) Quanto menor a variedade de informações contextuais, maior é a probabilidade da corretude da adaptação ser menor.</p> <p>1.2) A dependência a um determinado recurso (ex: aplicativo externo) pode afetar a corretude da adaptação, caso esse aplicativo não funcione</p> <p>1.3) Quanto menor a precisão da informação/situação de contexto, menor pode ser a corretude da adaptação</p> <p>1.4) Se o contexto muda constantemente, uma adaptação pode acontecer antes que uma mudança de contexto aconteça, resultando em uma adaptação incorreta</p> <p>4.1) A velocidade de conexão da rede pode variar o tempo de adaptação</p> <p>5.1) Dependendo da técnica escolhida, o grau de interferência será maior.</p>	

Figura 4.2 – *Abstract Sheet* para Sensibilidade ao Contexto

Para este foco, foi analisado o significado de um sistema ser sensível ao contexto, já apresentado na Seção 2.1.3. Assim, foi observado que muitos aspectos da sensibilidade ao contexto impactam na qualidade da interação e, portanto, precisam ser avaliados.

O primeiro aspecto é a corretude da adaptação, que significa verificar se o sistema se adapta de forma correta ao contexto real do usuário, provendo serviços e informações corretamente. Alguns fatores podem influenciar essa corretude, como, por exemplo: a própria corretude do contexto, pois se o contexto estiver errado, a adaptação provavelmente estará. Por exemplo, um aplicativo que possui adaptações de acordo com o nível de bateria. A coleta do nível de bateria deve estar correta, para que a adaptação seja realizada corretamente. Outro fator que foi levado em consideração: frequência de mudança do contexto, se a mudança de contexto ocorre frequentemente, a adaptação pode não ocorrer antes de ocorrer outra mudança de contexto. Esse fator foi observado no artigo de (CHENG et al., 2009).

Por fim, outro aspecto é o tempo que o sistema leva para adaptar, pois a informação e/ou serviços devem ser entregues em um tempo apropriado para o usuário. Todos esses aspectos e as informações extraídas das entrevistas foram levados em consideração para definição das questões e das medidas de *software* para responder a essas questões.

Essas informações obtidas nos *abstract sheets* podem ou não se transformar em medidas de *software*. Algumas delas podem apenas funcionar como um meio para interpretar os resultados encontrados pela coleta de dados. Por exemplo, a informação “Uso de técnicas para raciocinar sobre as incertezas de contexto”, pode auxiliar na interpretação de resultados provenientes de medidas como o grau de interferência do usuário, pois dependendo da técnica utilizada, mais ou menos interferências do usuário podem ser necessárias.

O segundo *abstract sheet* gerado foi para o foco de qualidade “*Mobilidade*” e está apresentado na Figura 4.3. Mobilidade na computação ubíqua significa romper todas as ligações entre o usuário, o dispositivo, a aplicação e o ambiente. Embora existam muitos trabalhos desenvolvidos para alcançar a mobilidade na computação ubíqua, é importante destacar que ainda é um assunto recente e que precisa de mais investigações (YU et al., 2013).

Esse *abstract sheet* apresenta uma preocupação dos especialistas com relação à capacidade do sistema de manter a interação com o serviço em diferentes dispositivos computacionais. Essa preocupação pode ser entendida através do seguinte cenário descrito por (YU et al., 2013):

Objeto	Propósito	Foco da Qualidade	Ponto de vista
Sistema Ubíquo	Avaliar	Mobilidade	Especialistas e Usuários
<b>Foco da qualidade</b>		<b>Fatores de variação</b>	
1)Manutenção da sessão e das interações passadas 2)Grau de funcionamento da rede: Número de falhas de conexão com a rede por tempo de uso da aplicação 3)Capacidade de <i>Handoff</i> (tempo de procedimento) 4)Suporte ao acesso do sistema através diversas tecnologias de rede ( <i>e.g.</i> , wifi, 3G, 4G, <i>bluetooth.</i> )		2)4)Modelo do dispositivo	
<b>Hipóteses de <i>Baseline</i></b>		<b>Impacto na Hipótese de <i>Baseline</i></b>	
1)Quanto mais alto, melhor 2)3)4)Quanto maior, melhor		2.1)4.1)Dispositivos móveis antigos terão menos tipos de acesso a rede e mais falhas de conexão podem ocorrer	

**Figura 4.3 - Abstract Sheet para Mobilidade**

“Bob está assistindo a um filme fantástico em seu computador em casa. Logo é hora de sair para o escritório, mas ele quer continuar o vídeo em seu *smartphone* enquanto ele está no metrô. A mobilidade da aplicação permite que Bob continue assistindo o filme exatamente de onde ele foi suspenso. Este cenário implica que a aplicação pode transferir uma tarefa para qualquer dispositivo disponível a partir do ponto em que a tarefa foi suspensa, de modo a fazer bom uso de vários recursos de *hardware* portáteis.”

Essa manutenção da interação em diferentes dispositivos deve ocorrer de forma suave e natural. O usuário não deve se sentir distraído e importunado, por exemplo, esperando um longo tempo para que o serviço volte a funcionar no novo dispositivo ou percebendo falhas na visualização do serviço por causa do tamanho diferente da tela do novo dispositivo.

Outro ponto comentado durante as entrevistas sobre mobilidade é a capacidade do dispositivo em trocar de rede, caso necessário, para que o serviço continue sendo entregue ao usuário, evitando falhas de conexão. Um detalhe importante disso é a duração que essa troca pode levar. Se o procedimento demora muito para acontecer, a rede pode ficar abaixo do limite aceitável para transmissão e a entrega do serviço pode ser descontinuada.

Outro fator é a identificação do número de tecnologias de acesso à rede que a aplicação pode utilizar. Quanto mais tecnologias possuir, melhor para a mobilidade, pois caso uma conexão venha a cair, outro tipo de conexão pode ser estabelecida.

É importante salientar que algumas dessas informações dependem da infraestrutura de rede, do sistema operacional e do dispositivo no qual a aplicação será executada.

O terceiro *abstract sheet* gerado foi para o foco de qualidade “*Transparência*” e está apresentado na Figura 4.4.

Objeto Sistema Ubíquo	Propósito Avaliar	Foco da Qualidade Transparência	Ponto de vista Especialistas e Usuários
<p><b>Foco da qualidade</b></p> <p>1) Grau de interação implícita: Nº de entradas implícitas/nº de entradas totais</p> <p>2) Tipo de comportamento adaptativo: <u>Comportamento não adaptativo</u>: sistema não possui entradas implícitas <u>Comportamento adaptativo transparente</u>: possui entradas do usuário e implícitas, mas o usuário não pode observar as entradas implícitas que influenciam o comportamento <u>Comportamento adaptativo não-transparente</u>: usuário observa as entradas implícitas, mas não é capaz de controlá-las <u>Comportamento adaptativo diverso</u>: o usuário é capaz de controlar algumas dessas entradas</p> <p>3) Grau de proatividade do sistema = nº de ações que o sistema consegue fazer pelo usuário (é necessário estabelecer o nº de ações totais que o usuário faria no sistema e quantas o sistema pode substituir)</p> <p>4) Número de ações que o usuário precisa fazer para inicializar a aplicação (inclui não somente os sensores)</p>		<p><b>Fatores de variação</b></p> <p>2) Nível de experiência dos usuários: leigos, experiência com computação, experiência com ubiquidade</p> <p>1)3) Decisões de projeto sobre as técnicas de implementação que influenciam na entrada implícita. Por exemplo, para localização: QRCode, NFC, Wifi, GPS, Fusão de sensores (Magnetômetro, acelerômetro)</p>	
<p><b>Hipóteses de <i>Baseline</i></b></p> <p>1) Quanto maior, mais transparente</p> <p>2) O que neste caso é melhor?</p> <p>3) Quanto maior o grau de proatividade, maior é a transparência</p> <p>4) Quanto menor, melhor</p> <p>5) Quanto maior, mais natural é a interação</p> <p>6) Quanto menor, melhor</p> <p>7) Quanto menos o usuário entender, mais transparente é para ele</p>		<p><b>Impacto na Hipótese de <i>Baseline</i></b></p> <p>2) Quanto mais experiente for o usuário, mais ele entenderá sobre o funcionamento da aplicação</p> <p>3) Dependendo da decisão de como será implementada a localização indoor, o grau de proatividade será menor</p>	

**Figura 4.4 - Abstract Sheet para Transparência**

Para a transparência, os especialistas se mostraram preocupados com o quanto o sistema é proativo no sentido de substituir tarefas dos usuários e projetar entradas implícitas, bem como o grau de esforço para colocar uma aplicação para funcionar, por exemplo: ligando sensores. Isso tudo resulta na percepção do usuário que o sistema existe, o que vai contra as aspirações da computação ubíqua.



Além disso, um especialista sugeriu verificar o tipo de comportamento adaptativo do sistema, o que significa verificar se usuário é capaz de entender as entradas dos sistemas e, com isso, se ele é capaz de controlá-las ou manipulá-las. Por exemplo, em uma casa inteligente, um dos serviços pode ser um detector de incêndios para disparar um sistema de água. Um usuário que consegue entender o que pode ser feito para manipular essas entradas pode, intencionalmente ou não, manipular uma adaptação.

O quarto *abstract sheet* gerado foi para o foco de qualidade “Atenção” e está apresentado na Figura 4.5 abaixo.

Objeto Interação	Propósito Avaliar	Foco da Qualidade Atenção	Ponto de vista Especialistas e Usuários
<p><b>Foco da qualidade</b></p> <p>1) Frequência de mudanças de foco: número de vezes que o usuário muda o foco devido a tecnologia pelo tempo total de uso</p> <p>2) Grau de interrupções Forma de cálculo: X-Y X = tempo de uso da aplicação Y = somatório das durações de todas as interrupções (entende-se por interrupção, aquelas causadas pela aplicação, ex: falhas) É o Percentual de tempo que o usuário gasta trocando entre dois focos</p> <p>3) Intensidade da Interrupção - Durações das interrupções - N° de vezes que interrupções aconteceram - Calcular desvio-padrão</p>		<p><b>Fatores de variação</b></p> <p>1.3) Quantidade de requisições de entrada 1.4) Número de erros/falhas 1.5) Duração de alertas sonoros/visuais que a aplicação utiliza</p>	
<p><b>Hipóteses de <i>Baseline</i></b></p> <p>1) Quanto maior, pior 2) Quanto maior o grau de interrupção, menor será o foco em outras atividades 3) Essa medida indica se durante o uso, houve várias interrupções, mas com pequenas durações ou poucas interrupções, mas com um tempo de duração maior Quanto menor, melhor</p>		<p><b>Impacto na Hipótese de <i>Baseline</i></b></p> <p>1) Quanto maior o 1.1, 1.2, 1.3, 1.4, 1.5, maior será a frequência de mudanças de foco</p>	

**Figura 4.5 - Abstract Sheet para Foco**

Para verificar o aumento da atenção do usuário nas suas tarefas principais, os especialistas sugeriram verificar a quantidade de vezes durante a interação que o usuário desvia a atenção da sua tarefa principal, como mencionado em (SCHOLTZ; CONSOLVO, 2004).

Essa mudança de foco é causada por uma ação que não está planejada como uma ação do sistema, por exemplo: ligar um determinado sensor para que a aplicação funcione. Isso retira o foco de atenção do usuário de sua atividade em andamento para realizar uma ação que não corresponde a uma entrada do sistema. Outros fatores que causam a mudança do foco podem ser a quantidade de requisições de entrada explícitas e o número de falhas. Caso a quantidade de requisições explícitas seja alta, o usuário deverá constantemente parar sua atividade para interagir diretamente com o sistema.

A *Atenção* está fortemente ligada com a característica de transparência, se um sistema é transparente para o usuário, provavelmente a sua atenção nas tarefas principais será maior.

Na Figura 4.6, é apresentado o sexto e último *abstract sheet* gerado nas entrevistas GQM.

<b>Objeto</b> Interação	<b>Propósito</b> Avaliar	<b>Foco da Qualidade</b> Interação Calma	<b>Ponto de vista</b> Especialistas e Usuários
<b>Foco da qualidade</b>		<b>Fatores de variação</b>	
<p>1) <u>Grau de Disponibilidade</u>: Alta, média, baixa e muito baixa</p> <p>2) <u>Grau de relevância da interação</u>: Alta, média, baixa e muito baixa</p> <p>3) <u>Grau de sincronia da interação</u>: Alta, média, baixa e muito baixa</p> <p>4) <u>Grau de Cortesia da interação</u>: Alta, média, baixa e muito baixa</p> <p>5) Grau de acerto entre a ação do sistema e a ação que o usuário requisitaria = quantidade de vezes que o sistema executa ações iguais as que o usuário executaria</p> <p>6) Grau de suporte à interações naturais = n° de tipos de interfaces naturais que a aplicação utiliza (baseada em voz, tangível, gestos)</p>		<p>Variedade de informações de contexto do usuário suportadas pela aplicação (Por exemplo, se o usuário estiver dirigindo, a aplicação não pode entregar muitas informações visuais, mas pode entregar de outra maneira, por voz, por exemplo.)</p> <p>Erros de informações coletadas, profundidade das informações, quantidade das informações coletadas</p>	
<b>Hipóteses de <i>Baseline</i></b>		<b>Impacto na Hipótese de <i>Baseline</i></b>	
<p>1)2)3)4)5) Quanto mais alto o valor, melhor</p> <p>2) Quanto menor, melhor</p> <p>4) Quanto mais preciso melhor</p>		<p>Quanto menos erros de informações coletadas e quanto maior o conhecimento de informações contextuais sobre as atividades do usuário, maior será o grau de acerto</p>	

**Figura 4.6-** *Abstract Sheet* para Interação Calma

Para o foco de qualidade “calma”, nenhuma medida foi encontrada pelo MS. No entanto, foi encontrado um estudo (RIEKKI; ISOMURSU; ISOMURSU, 2004) que propõe um *framework* conceitual que objetiva avaliar uma aplicação calma. Nesse artigo, o autor afirma que calma está

relacionado com quatro fatores: disponibilidade, relevância da interação, *timing* da sensibilidade ao contexto e cortesia da interação. Além disso, um importante ponto a se considerar em calma é que quando a aplicação precisa interagir com o usuário, essa interação deve acontecer da melhor forma possível, por exemplo, a aplicação pode reconhecer que o usuário está dirigindo e assim interagir baseado em comandos de voz. O importante é que quando o usuário tiver que ser interrompido, que seja da melhor forma possível.

No entanto, calma é uma característica vivenciada pelo usuário e somente pode ser avaliada por ele em um contexto real de uso. Um sistema pode ser calmo para um usuário, mas para outro usuário não. (RIEKKI; ISOMURSU; ISOMURSU, 2004) propõem fazer a avaliação de forma subjetiva definindo valores como: alto, médio, baixo e muito baixo para cada um dos fatores definidos. Esses valores servirão como elementos de medição para o foco “calma”.

Finalizadas as entrevistas sobre as cinco características, questões e medidas de *software* foram definidas conforme as informações obtidas. As próximas subseções apresentam as definições de cada uma delas.

#### 4.3.4 Questões

Com base nas informações obtidas nas entrevistas e documentadas nos *abstracts sheets*, foram derivadas as seguintes questões do GQM:

- **Q1** - Qual o grau de corretude da adaptação sensível ao contexto?
- **Q2** - Qual o grau de corretude do contexto?
- **Q3** - Qual o grau de suporte à mobilidade?
- **Q4** - Qual o grau de transparência da aplicação?
- **Q5** - Qual o grau de requisição da atenção do usuário?
- **Q6** - Qual o grau de calma da interação?

A questão 1 (“Qual o grau de corretude da adaptação sensível ao contexto?”) busca identificar se as adaptações sensíveis ao contexto estão acontecendo de forma correta para o usuário, isso significa identificar se o sistema provê serviços e informações corretos de acordo com o contexto real do usuário, bem como em tempo hábil. Por exemplo, o usuário está dirigindo e necessita que um determinado sistema entregue o serviço através de uma interface baseada em voz, o sistema deve ser capaz de adaptar essa interface no momento em que o usuário está dirigindo.

Existe a necessidade, então, de verificar se o sistema consegue capturar corretamente o contexto real do usuário, o que significa verificar se o contexto coletado corresponde com a situação real do usuário. Por exemplo: o usuário está realmente dirigindo? Alguns fatores que influenciam essa corretude também devem ser medidos, como a granularidade da informação de contexto. Assim, foi derivada a questão 2 (“Qual o grau de corretude do contexto?”), portanto, todas as medidas que serão apresentadas para essa questão, também podem auxiliar na interpretação das respostas à questão 1.

A questão 3 (“Qual o grau de suporte à mobilidade?”) busca identificar se existe um bom suporte ao sistema para a mobilidade, de forma que o usuário não se sinta importunado por desconexões ou descontinuação dos serviços.

A questão 4 (“Qual o grau de transparência da aplicação?”) busca identificar o grau de interação transparente entre o usuário e o sistema. Além disso, é importante saber se o usuário tem conhecimento de que as entradas do sistema podem ser manipuladas. Quanto mais o usuário conhecer sobre isso, menos transparente o sistema será para ele e isso pode depender da experiência do usuário com a computação.

A questão 5 (“Qual o grau de requisição da atenção do usuário?”) busca identificar se o usuário aumentou seu foco de atenção nas atividades cotidianas. Isso pode ser respondido através de medidas como: a quantidade de vezes, durante a interação, que o usuário desvia a atenção da sua tarefa principal; a quantidade de requisições de entrada explícitas; e o número de falhas. Se durante a interação, o usuário teve um alto grau de distrações, então a atenção nas suas atividades cotidianas está baixa.

A questão 6 (“Qual o grau de calma da interação?”) está relacionada com o quanto a interação é calma do ponto de vista do usuário. Como, por exemplo, se o sistema está disponível quando necessário ou se não aborrece os usuários com informações ou requisições não relevantes. Além disso, é verificado se o sistema entrega os serviços no momento correto e da melhor maneira possível de acordo com a situação atual do usuário.

Cada uma dessas questões vai ser respondida por meio da definição de medidas de *software*. Tais medidas são apresentadas na próxima subseção.

### **4.3.5 Medidas**

Para responder as questões definidas na Seção 4.3.4, um conjunto de medidas de *software* foi especificado com base nas contribuições dos especialistas durante as entrevistas e, também, nas medidas coletadas na literatura através do MS (Apêndice A).

Cada medida foi definida de acordo com o formato da (ISO/IEC 25000, 2005). O formato da ISO contém as seguintes informações: um ID, um nome, uma descrição, uma função de medição e seus elementos de medição de qualidade e um método de coleta. A princípio a norma não define valores de interpretação, pois todas as medidas aplicam suas funções de medição para resultarem entre 0.0 e 1.0 e assim quanto mais perto de 1.0, melhor. Apenas em alguns casos, a interpretação é diferente e então ela é descrita na nota da medida. Como neste, trabalho nem todos os valores de interpretação são os mesmos, uma coluna de “interpretação” foi adicionada no formato de descrição das medidas.

O ID definido para as medidas desse trabalho seguem o seguinte formato: MX.Y, onde X é o número da questão do GQM que ela responde e Y corresponde ao número da medida. As medidas são referenciadas ao longo do texto através de seus IDs.

Os métodos de coleta para essas medidas foram definidos como: (i) observação: na qual um avaliador observa o comportamento da aplicação; (ii) automática: no qual a aplicação é instrumentada e logs de interação são gerados; (iii) questionário do usuário: onde o usuário é questionado através de formulários; e (iv) questionário do desenvolvedor do sistema: no qual o desenvolvedor responde ao questionário de avaliação.

Além dessas informações, o formato de documentação das medidas desse trabalho possui uma coluna com “referência”. Essa coluna indica que a medida pode ter sido retirada ou baseada em medidas encontradas pelo MS (neste caso, elas são referenciadas pelo ID MX, ver Apêndice A) ou de medidas presentes em outros trabalhos que não saíram no mapeamento, mas são de conhecimento dos especialistas que participaram das entrevistas.

A Tabela 4.2 apresenta o conjunto de medidas para responder à questão 1 (“Qual o grau de corretude das adaptações sensíveis ao contexto?”). Neste caso, foram definidas quatro medidas de *software*, as quais avaliarão a corretude das adaptações ocorridas durante o uso da aplicação pelos usuários.

**Tabela 4.2** – Medidas de Software para a Questão 1

ID	Nome da Medida	Descrição	EMQs e Funções de Medição	Interpretação	Método de Coleta	Ref
M1.1	Corretude da Adaptação	A adaptação ocorreu de maneira correta no contexto atual do usuário?	$X = \frac{\left(\sum_{i=1}^N \frac{A_i}{B_i}\right) * 100}{N}$ <p> <math>A_i</math> = nº de adaptações <math>i</math> executadas corretamente  <math>B_i</math> = nº total de adaptações <math>i</math> executadas  <math>N</math> = nº dos tipos das diferentes adaptações                 </p>	Quanto mais próximo de 100%, melhor.	Automática e Observação	Baseada em M2
M1.2	Intervalo de mudanças do contexto	Qual a frequência de mudança do contexto?	$\mathbf{X} = \mathbf{F}$ , onde F é a média dos intervalos de tempo entre uma mudança e outra.	Baixa = minutos Média = segundos Alta = milissegundos	Automática	Retirada de (CHENG et al., 2009)
M1.3	Tempo de Adaptação	Quanto tempo leva para a adaptação ocorrer?	$\mathbf{X} = \mathbf{T}$ , onde T é a média dos tempos que o sistema demorou em se adaptar uma vez que mudou o contexto	Baixo = milissegundos Médio = segundos Alto = minutos	Automática	Retirada de (CHENG et al., 2009)
M1.4	Grau de Adaptação	As adaptações sempre ocorrem quando necessário?	$X = \frac{\left(\sum_{j=1}^N \frac{A_j}{B_j}\right) * 100}{N}$ <p> <math>A_j</math> = nº de vezes que adaptou  <math>B_j</math> = nº de vezes que foi requisitada a adaptação (mudança contextual)  <math>N</math> = nº dos tipos das diferentes adaptações                 </p>	Quanto mais próximo de 100%, melhor.	Automática e Observação	Baseada em M2

A medida M1.1 busca identificar se a adaptação ocorreu de maneira esperada para o usuário. Isso significa verificar se os serviços e/ou as informações foram entregues de maneira correta em relação ao que é esperado por ele no momento atual. Por exemplo, se o contexto atual do usuário é “dirigindo”, então uma adaptação para entregar o serviço através de uma interface baseada em voz deve ser feita. Assim, essa medida verifica se o serviço realmente utilizou a interface baseado em voz. Para calcular essa medida, é necessário identificar quais adaptações o sistema possui e que podem ser coletadas. Assim, é possível contar quantas vezes uma determinada adaptação identificada ocorreu durante um determinado período de uso da aplicação e quantas dessas vezes ocorreram corretamente. Como um sistema pode ter várias adaptações, a medida apresenta um somatório com o objetivo de somar todas as adaptações ocorridas, para depois calcular uma média ao dividir por N.

A medida M1.2 mede a periodicidade em que uma informação de contexto é coletada para ser utilizada para executar uma adaptação. É muito importante medir isso, pois, conforme já mencionado, se a frequência de mudança no contexto é alta, uma adaptação pode acontecer antes que uma mudança de contexto aconteça, resultando em uma adaptação incorreta do ponto de vista do usuário. Para calculá-la é necessário registrar os horários em que mudanças de contexto ocorrem e contar o tempo entre essas mudanças contextuais. Por fim, uma média de todos os intervalos de tempo calculados deve ser feita para obter um número total para essa medida.

A medida M1.3 mede o tempo que o sistema leva para se adaptar a partir do instante em que a informação de contexto foi coletada ou detectada. Essa medida é importante porque a informação e/ou o serviço deve ser entregue em um tempo apropriado para o usuário. Para calcular o resultado dessa medida, é necessário calcular uma média entre os tempos de adaptação. Na interpretação dessa medida, também pode ser verificado se houve resultados muito distantes uns dos outros, através do cálculo do desvio-padrão.

A medida M1.4 busca verificar se as adaptações ocorreram quando requisitadas por uma mudança contextual. Essa medida foi definida porque pode acontecer do sistema não conseguir se adaptar mesmo acontecendo uma mudança contextual. Para calcular essa medida, primeiro, é necessário identificar quais adaptações e mudanças de contexto o sistema possui e que podem ser coletadas. Segundo, é necessário contar quantas vezes uma determinada mudança contextual (requisição para a adaptação ocorrer) aconteceu durante um determinado período de uso da aplicação e quantas dessas vezes ocorreram a adaptação. Como um sistema pode ter várias adaptações, a medida apresenta um somatório com o objetivo de somar todas as adaptações ocorridas, para depois calcular uma média ao dividir por N. A próxima Tabela 4.3 apresenta o conjunto de medidas para responder à questão 2.

Tabela 4.3 - Medidas de Software para a Questão 2

ID	Nome da Medida	Descrição	EMQs e Funções de Medição	Interpretação	Método de Coleta	Ref
M2.1	Corretude do contexto	As informações de contexto correspondem ao contexto atual do usuário?	$X = \frac{\left(\sum_{i=1}^N \frac{A_i}{B_i}\right) * 100}{N}$ <p> <math>A_i</math> = nº de informações de contexto <math>i</math> corretas coletadas  <math>B_i</math> = nº de informações de contexto <math>i</math> coletadas  <math>N</math> = nº das diferentes informações de contexto                 </p>	Quanto mais próximo de 100%, melhor.	Automática e Observação	Baseada em M23
M2.2	Corretude das situações de contexto	As situações de contexto correspondem às situações atuais dos usuários?	$X = \frac{\left(\sum_{j=1}^N \frac{A_j}{B_j}\right) * 100}{N}$ <p> <math>A_j</math> = nº de situações de contexto <math>j</math> coletadas corretamente  <math>B_j</math> = nº de situações <math>i</math> coletadas  <math>N</math> = nº das diferentes situações de contexto                 </p>	Quanto mais próximo de 100%, melhor.	Automática e Observação	Baseada em M23
M2.3	Granularidade do Contexto	Qual a granularidade do contexto?	<p><b>X=A</b>, onde A pode ser,</p> <p>(1) Baixa: Informações de Contexto de Baixo Nível</p> <p>(2) Média: Situações de Contexto</p> <p>(3) Alta: Informações de Alto Nível</p>	<p>(1) Baixa: Informações de Contexto de Baixo Nível</p> <p>(2) Média: Situações de Contexto</p> <p>(3) Alta: Informações de Alto Nível</p>	Questionário do Desenvolvedor	Retirada de M26 e



As medidas M2.1 e M2.2 buscam identificar se as informações e/ou situações de contexto capturadas pelo sistema espelham a realidade. Para calcular essas medidas, é necessário identificar quais informações ou situações de contexto o sistema utiliza para executar adaptações e quais podem ser coletadas. Sua função de medição corresponde a contar quantas vezes uma informação ou situação de contexto foi coletada e quantas dessas vezes a informação estava correta do ponto de vista do usuário. Isso significa avaliar se a informação de contexto ou situação corresponde à situação atual do usuário. Por exemplo, uma situação “*em\_aula*” é considerada correta se realmente o usuário estiver em aula. Para coletar esses dados, um *log* dos contextos capturados deve ser gerado, bem como a observação do avaliador ou do próprio usuário sobre o contexto em que ele se encontrava. Para completar o cálculo dessa medida, a função de medição apresenta um somatório, pois um sistema pode ter várias informações de contexto e/ou situações sendo capturadas, com o objetivo de somar todas as informações/situações de contexto coletadas para depois calcular uma média ao dividir por N.

É importante salientar que a medida M2.1 identifica se as informações de contexto estão corretas. Entretanto, pode ser que as informações de contexto coletadas não representem exatamente a realidade, porém, por estarem muito próximas desta, podem ser consideradas corretas. Por exemplo, um GPS captura a localização de um carro com uma diferença de 2 metros para a realidade. No entanto, o desenvolvedor pode julgar essa diferença muito insignificante e considerar que a informação de contexto está correta.

A medida M2.3 avalia a granularidade do contexto utilizado. Essa medida foi retirada do trabalho de (BETTINI et al., 2010). Como apresentado na Seção 2.1.3, aplicações que utilizam somente informações de contexto de baixo nível podem ser sem sentido e vulneráveis a pequenas mudanças ou incertezas, influenciando assim a corretude das adaptações. Desta forma, as aplicações que utilizam essas informações de contexto podem ser de pouca utilidade. Quanto maior o nível de tratamento das informações de contexto, maior será a probabilidade das adaptações estarem corretas.

Todas as medidas definidas para essa segunda questão podem ser reutilizadas para responder a primeira questão sobre corretude das adaptações, pois estas dependem fortemente da corretude dos contextos coletados.

Para a questão 3, foi definido um conjunto com quatro medidas de software, apresentado na Tabela 4.4.

**Tabela 4.4** – Medidas de Software para a Questão 3

ID	Nome da Medida	Descrição	EMQs e Função de Medição	Interpretação	Método de Coleta	Ref
M3.1	Indicador de Mobilidade Transparente	O sistema é capaz de se mover entre diferentes dispositivos de forma suave e discreta?	$X = A$ , onde (0) Inexistente (1) Baixo (2) Médio (3) Alto	(0) Inexistente (1) Baixo (2) Médio (3) Alto	Observador	Retirada de (YU et al., 2013)
M3.2	Grau de Funcionamento da Rede	Quantas falhas de conexão ocorrem?	$x = N^{\circ}$ total de falhas de conexão ocorridas	Quanto mais perto de 0, melhor.	Automática e Observação	Retirada de (ISO/IEC 9126, 2001)
M3.3	Duração do Procedimento de <i>Handoff</i>	Quanto tempo demora para realizar a troca de rede?	$X = T$ , onde T é o tempo que leva para executar a troca da rede	Quanto mais próximo de 0, melhor.	Automática	Retirada de (SGORA; VERGADOS, 2009)
M3.4	Suporte ao acesso do sistema	Quantas tecnologias de acesso à rede o sistema possui?	$X = A$ , onde $A = N^{\circ}$ de tecnologias de acesso à rede	Quanto maior, melhor	Questionário do Avaliador	Baseada em M54

A medida M3.1 busca identificar a capacidade da aplicação de dar suporte ao acesso do usuário em diferentes dispositivos e ainda manter suas interações e sessões passadas com um mínimo de distrações possíveis.

Logo, essa medida busca identificar exatamente isso, se é possível manter a sessão passada exatamente de onde o usuário parou em outro dispositivo móvel da melhor maneira possível. Para defini-la foram utilizados as informações contidas no trabalho de (YU et al., 2013) sobre mobilidade:

- Alto é quando a aplicação consegue se mover de um dispositivo para outro, mantendo as interações passadas e se adaptando aos recursos do novo dispositivo (por exemplo: tamanho da tela), de forma que o usuário continua suas tarefas assim que troca de dispositivo.
- Médio é quando a aplicação consegue se mover de um dispositivo para outro, mantendo as interações passadas e se adaptando aos novos recursos. No entanto, o usuário pode vir a ter que esperar muito para começar a interagir com a aplicação.
- Baixo é quando a aplicação consegue se mover para outro dispositivo. No entanto, a aplicação não se adapta aos novos recursos.
- Inexistente é quando a aplicação não consegue se mover de um dispositivo para outro.

A medida M3.2 está relacionada com quantas vezes o sistema fica sem conexão, assim impossibilitando a provisão dos serviços para os usuários. Essa medida pode ser coletada através da monitoração constante das conexões do dispositivo.

A medida M3.3 busca identificar a duração do procedimento de troca de redes. Esse processo se chama *handoff* e representa um elemento essencial para apoiar a mobilidade do usuário. *Handoff* é a capacidade de um dispositivo se conectar em uma rede quando sai da cobertura da rede na qual estava conectado, sem que o usuário perceba. Podem existir dois tipos de *handoff*: vertical e horizontal. O *handoff* horizontal é quando o dispositivo se move de uma rede para outra do mesmo tipo (por exemplo, duas redes *wifi*) e o *handoff* vertical acontece quando o dispositivo se move de um tipo de rede para outro (por exemplo, de *wifi* para 3G) (SGORA; VERGADOS, 2009).

Outra classificação para o *handoff* é que ele pode ser *hard* ou *soft*. *Hard Handoff* é aquele onde o sistema somente se conecta com uma nova rede após a conexão tiver sido quebrada e é geralmente referenciada por “*break before make*”. O *Soft Handoff* não espera a conexão quebrar, ele

se conecta antes disso e é referenciada como “*make before break*” Outras medidas de software para avaliar a qualidade do *handoff* podem ser encontradas em (SGORA; VERGADOS, 2009).

Essa medida não foi coletada nos estudos de caso porque não foi possível realizar uma medição precisa desse processo, uma vez que depende do Sistema Operacional em que aplicação está executando.

A medida M3.4 é calculada através da contagem do número de tecnologias de acesso à rede que a aplicação pode utilizar. Quanto mais tecnologias possuir, melhor para a mobilidade, pois caso uma conexão venha a cair, outro tipo de conexão pode ser estabelecida. Neste caso, essa medida depende do dispositivo no qual a aplicação está sendo desenvolvida. Diferentes dispositivos possuem diferentes interfaces de acesso a rede.

Para a questão 4, foi definido um conjunto com cinco medidas de *software*, apresentado na Tabela 4.5.

Tabela 4.5 –Medidas de software para a Questão 4

ID	Nome da Medida	Descrição	EMQs e Funções de Medição	Interpretação	Método de Coleta	Ref
M4.1	Grau de interação implícita	Qual o grau de interação implícita?	$X = (A / N) * 100$ , onde A = N° de entradas implícitas N = N° total de entradas da aplicação	Quanto mais próximo de 100%, melhor	Questionário do Desenvolvedor e Observação	Baseada em (BROY et al., 2009)
M4.2	Tipo de comportamento adaptativo	Qual o tipo de comportamento do usuário da aplicação?	$X = A$ , onde A pode ser (1) Comportamento não adaptativo (2) Comportamento adaptativo não-transparente (3) Comportamento adaptativo dissuadido (4) Comportamento adaptativo transparente	(1) Comportamento não adaptativo (2) Comportamento adaptativo não-transparente (3) Comportamento adaptativo dissuadido (4) Comportamento adaptativo transparente	Questionário do Desenvolvedor e Observação	Retirada de (BROY et al., 2009)
M4.3	Proatividade da aplicação	Quanto a aplicação pode substituir ações do usuário?	$X = N - A$ , onde N = N° de ações totais desenvolvidas que podem ser substituídas por sensores A = N° de ações a aplicação substitui	Quanto mais próximo de 0, melhor	Questionário do Desenvolvedor e Observação	Baseada em M3
M4.4	Grau de esforço para inicializar o sistema	Qual o grau de esforço para inicializar a aplicação?	$X = A$ , onde A = Número de ações que o usuário precisa fazer para inicializar a aplicação (inclui não somente manipular os sensores)	Quanto mais próximo de 0, melhor	Questionário do Desenvolvedor e Observação	Baseada em M34
M4.5	Grau de intervenção do usuário	Quanto o usuário precisa intervir para a adaptação ocorrer?	$X = \frac{\left(\sum_{i=1}^N \frac{A_i}{B_i}\right) * 100}{N}$ Ai = Quantidade de adaptações que o usuário precisa intervir explicitamente Bi = Quantidade total de adaptações executadas N = N° das diferentes adaptações	Quanto mais próximo de 100%, pior	Automática, Observação e Questionário do Desenvolvedor	Baseada em M17

A medida M4.1 busca identificar se o sistema foi projetado para utilizar entradas implícitas. Essas são entradas que não necessariamente são consideradas como entradas para o usuário e podem não ser observadas por este. Isso tornará a interação com um sistema computacional mais transparente, dado que o usuário não precisará interagir explicitamente. Para contabilizar essa medida, basta contar quantas entradas implícitas a aplicação possui e dividir pelo número total de entradas da aplicação, o que abrange o número de entradas implícitas e o número de entradas explícitas (medida definida para o foco de qualidade *Atenção*). O resultado é multiplicado por 100 para obter uma porcentagem, assim, quanto mais próximo o resultado estiver de 100%, mais próximo está o número de entradas implícitas do número de entradas total do sistema, tornando maior o grau de interação implícita.

A medida M4.2 foi baseada na classificação encontrada no trabalho de (BROY et al., 2009). As definições dos valores que essa medida pode assumir, nesse trabalho, são: (1) *Comportamento não adaptativo* é aquele que o comportamento do sistema é unicamente determinado pelo usuário, (2) *Comportamento adaptativo não transparente*, é aquele que o sistema possui entradas implícitas e o usuário é capaz de conhecer quais são e, assim, inferir um relacionamento entre os eventos que ocorrem no ambiente e as reações resultantes do sistema, (3) *Comportamento adaptativo dissuadido* é aquele que o usuário conhece as entradas implícitas e consegue controlá-las de maneira intencional ou não e (4) *Comportamento adaptativo transparente*, é aquele que o usuário não conhece as entradas implícitas que influenciam o ambiente e, portanto, não sabe como controlá-las, assim a interação com essa aplicação é totalmente transparente. Ao calcular essa medida, é necessário levar em consideração a experiência do usuário com relação à computação. Obviamente, os usuários que possuem experiência com computação, principalmente, computação sensível ao contexto, serão capazes de observar entradas implícitas e, caso possível, manipulá-las, tornando, assim, o tipo de comportamento não transparente ou dissuadido. Por outro lado, os usuários que não possuem essa experiência, o tipo de comportamento será transparente.

A medida M4.3 busca conhecer qual o grau de proatividade da aplicação ao contabilizar quantas ações do usuário, a aplicação é capaz de substituir. Essa medida é contabilizada através da contagem de quantas ações o usuário faria na aplicação se ela não utilizasse nenhum tipo de sensor para capturar informações e quantas dessas, a aplicação substitui. Quanto mais próximo o resultado for de

A medida M4.4 busca identificar qual esforço necessário para o usuário utilizar uma aplicação. Por exemplo, o usuário pode necessitar que um sensor seja ligado para que a aplicação

possa funcionar. Essa ação deixa o usuário ciente da computação, o que vai contra as aspirações da computação ubíqua.

A medida M4.5 busca identificar se as adaptações estão ocorrendo de um modo pelo qual não é preciso o usuário intervir para que ela aconteça. Essa medida foi derivada do fato de que incertezas de contexto podem acontecer. Incertezas podem ser geradas de três formas: contexto incerto, informações ambíguas, contexto errado (DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011). Uma das formas de tratar isso é perguntando ao usuário sobre a correte de um determinado contexto, caso isso aconteça muitas vezes, o usuário se sentirá importunado e provavelmente deixará de utilizar a aplicação por não sentir confiança nela.

Além desse ponto, em determinadas aplicações, a adaptação pode ocorrer somente se o usuário intervir explicitamente fornecendo uma entrada para a aplicação. Essa medida é calculada verificando quantas adaptações o usuário precisou intervir pela quantidade total de adaptações que um sistema realizou. Existem técnicas para tratar incertezas de contexto (Lógica Fuzzy, Lógica Probabilística, Redes Bayesianas, Modelos Oculto de Markov, Teoria de Dempster-Shafer, Requisição para o usuário) (BETTINI et al., 2010). O uso dessas técnicas pode levar a uma minimização nas intervenções do usuário.

Essa medida (M4.5) pode ser reaproveitada para analisar a questão 2 (correte de contexto), pois se o usuário precisa intervir por motivos de conflito de contexto pode ser porque a correte de contexto não está em um grau apropriado.

A medida M3.1 (Indicador de Mobilidade Transparente), já apresentada e discutida na questão 3, é reaproveitada para esta questão pois trata de transparência na troca de dispositivos móveis pelos usuários.

Para a questão 5, foi definido um conjunto com 4 medidas, apresentado na Tabela 4.6.

**Tabela 4.6** – Medidas de Software para a Questão 5

<b>ID</b>	<b>Nome da Medida</b>	<b>Descrição</b>	<b>EMQs e Funções de Medição</b>	<b>Interpretação</b>	<b>Método de Coleta</b>	<b>Ref</b>
M5.1	Quantidade de mudanças de foco	Quantas mudanças de foco do usuário de suas atividades acontecem ao longo do uso da aplicação?	$X = A$ , onde $A = N^{\circ}$ de ações que mudam o foco do usuário durante o uso	Quanto mais longe de 0, melhor.	Observação e Automática	Baseada em M46, M49
M5.2	Grau reconhecido de tempo perdido durante o uso	Qual a quantidade de tempo perdido durante o uso da aplicação?	$X = B$ , onde $B =$ Somatório dos tempos perdidos	Quanto mais longe do tempo de utilização, melhor	Automática	Baseada em M7
M5.3	Número de Entradas explícitas	Qual a quantidade de entradas explícitas do usuário?	$X = A$ , onde $A = N^{\circ}$ de requisições de entrada do usuário	Quanto mais perto de 0, melhor	Observação	Baseada em M43
M5.4	Tempo Médio Entre Falhas	Quão frequentemente falhas ocorrem?	$X = T/A$ , onde $T =$ Tempo de operação $A = N^{\circ}$ de falhas ocorridas	Quanto mais longe de 0, melhor.	Automática e Observação	Retirada de (ISO/IEC 9126, 2001)



A medida M5.1 busca identificar a quantidade de vezes que o usuário teve que mudar o foco devido a tecnologia. Essa mudança de foco é causada quando alguma ação do usuário deve ser realizada para que a aplicação funcione durante o uso. Por exemplo, reiniciar um determinado sensor para que a aplicação funcione durante o uso e/ou reiniciar a aplicação são ações que mudam o foco de atenção do usuário para a aplicação desnecessariamente. Essa medida é coletada através de logs da interação e também observando a utilização da aplicação pelo usuário e verificando quantas e quais ações foram requisitadas para o usuário que não fazem parte da aplicação. Quanto maior é o número de ações que mudam o foco do usuário, maior será a frequência de mudança do foco da atenção.

A medida M5.2 busca identificar a quantidade de tempo total que o usuário perdeu, como por exemplo: esperando uma adaptação ocorrer, uma saída ser exibida, requisição de entrada explícita (digitação). Assim é possível saber qual foi o tempo de uso da aplicação sem o tempo perdido, diminuindo esse valor do valor do tempo total de uso do sistema.

A medida M5.3 busca identificar a quantidade de entradas explícitas do usuário, caso esse número seja muito grande, o grau de requisição da atenção do usuário será maior.

A medida M5.4 tem como objetivo identificar a frequência com que o sistema falha em operação. Esses dados podem ser obtidos durante os testes, por logs e/ou observação da interação. É necessário contar o número de falhas ocorridas durante um período de operação. Quanto mais distante de zero for o resultado dessa medida, melhor, pois significa que demora para uma falha ocorrer durante operação do sistema.

A medida M3.2 foi especificada para a mobilidade, mas também deve ser utilizada para essa questão, pois está relacionado com quantas vezes o sistema fica sem conexão, impossibilitando assim, a provisão dos serviços para os usuários, e assim tirando a atenção do usuário de suas atividades.

A medida M4.5 (Grau de intervenção do usuário) também pode ser reaproveitada para esta questão. Se durante a interação o usuário é muito requisitado para que as adaptações ocorram, então a atenção dele de suas atividades cotidianas está diminuindo.

A Tabela 4.7 apresenta o conjunto de medidas para responder à questão **6**, neste caso foram definidas quatro medidas de software, as quais avaliam o grau de calma da interação.

**Tabela 4.7 - Medidas de software para a Questão 6**

ID	Nome da Medida	Descrição	EMQs e Funções de Medição	Interpretação	Método de Coleta	Ref
M6.1	Grau de disponibilidade	A aplicação está disponível para o usuário quando ele necessita?	$X = A$ , onde A é (1)Alto (2)Médio (3)Baixo (4)Muito baixo	(1)Alto (2)Médio (3)Baixo (4)Muito baixo	Questionário do Usuário	Retirada de (RIEKKI; ISOMURSU; ISOMURSU, 2004)
M6.2	Grau de relevância da interação	A aplicação fornece somente informações relevantes?	$X = A$ , onde A é (1)Alto (2)Médio (3)Baixo (4)Muito baixo	(1)Alto (2)Médio (3)Baixo (4)Muito baixo	Questionário do Usuário	Retirada de (RIEKKI; ISOMURSU; ISOMURSU, 2004)
M6.3	Grau de sincronia da interação	A aplicação interage no momento correto?	$X = A$ , onde A é (1)Alto (2)Médio (3)Baixo (4)Muito baixo	(1)Alto (2)Médio (3)Baixo (4)Muito baixo	Questionário do Usuário	Retirada de (RIEKKI; ISOMURSU; ISOMURSU, 2004)
M6.4	Grau de cortesia da interação	A interação ocorre da melhor maneira possível?	$X = A$ , onde A é (1)Alto (2)Médio (3)Baixo (4)Muito baixo	(1)Alto (2)Médio (3)Baixo (4)Muito baixo	Questionário do Usuário	Retirada de (RIEKKI; ISOMURSU; ISOMURSU, 2004)

Como mencionado anteriormente, a interação calma deve ser medida de acordo com a percepção do usuário, então foram utilizados os fatores e valores propostos por (RIEKKI; ISOMURSU; ISOMURSU, 2004). Para a medida M6.1, é necessário seguir as seguintes direções especificadas pelos autores para cada valor:

- Alto: praticamente sempre disponível para o usuário quando ele necessita. Quando uma falha ou condição inesperada ocorre, o usuário pode executar normalmente a função desejada sem nenhuma dificuldade ou perturbação.
- Médio: Geralmente está disponível quando necessita, quando uma falha ou condição inesperada acontece, o usuário pode executar a aplicação, mas com dificuldades.
- Baixo: Disponível somente em alguns locais desejados. Possui restrições de localização, tempo e outras e quando uma falta ou condição inesperada surge, o usuário consegue executar a aplicação com muita dificuldade.
- Muito baixo: A disponibilidade não corresponde com o desejo do usuário de onde e quando utilizá-la. Quando uma falta ou condição inesperada ocorre, o usuário não é capaz de executar a função desejada.

Para a medida M6.2, é necessário seguir as seguintes direções:

- Alta: nenhuma informação, requisição e/ou escolha irrelevantes são apresentadas, o usuário obtém somente a informação necessária.
- Média: a maioria das informações e escolhas é relevante.
- Baixa: informações e requisições irrelevantes são frequentemente apresentadas para o usuário.
- Muito Baixa: muita informação é apresentada. Muitas opções de comando irrelevantes são apresentadas. O usuário precisa procurar comandos ou opções corretas em hierarquias de *menu* ou em uma longa lista de opções. O usuário sente que o seu contexto não tem nenhum efeito sobre a informação ou escolhas apresentadas. Falhas de adaptação também influenciam neste valor.

Para a medida M6.3, é necessário seguir as seguintes direções:

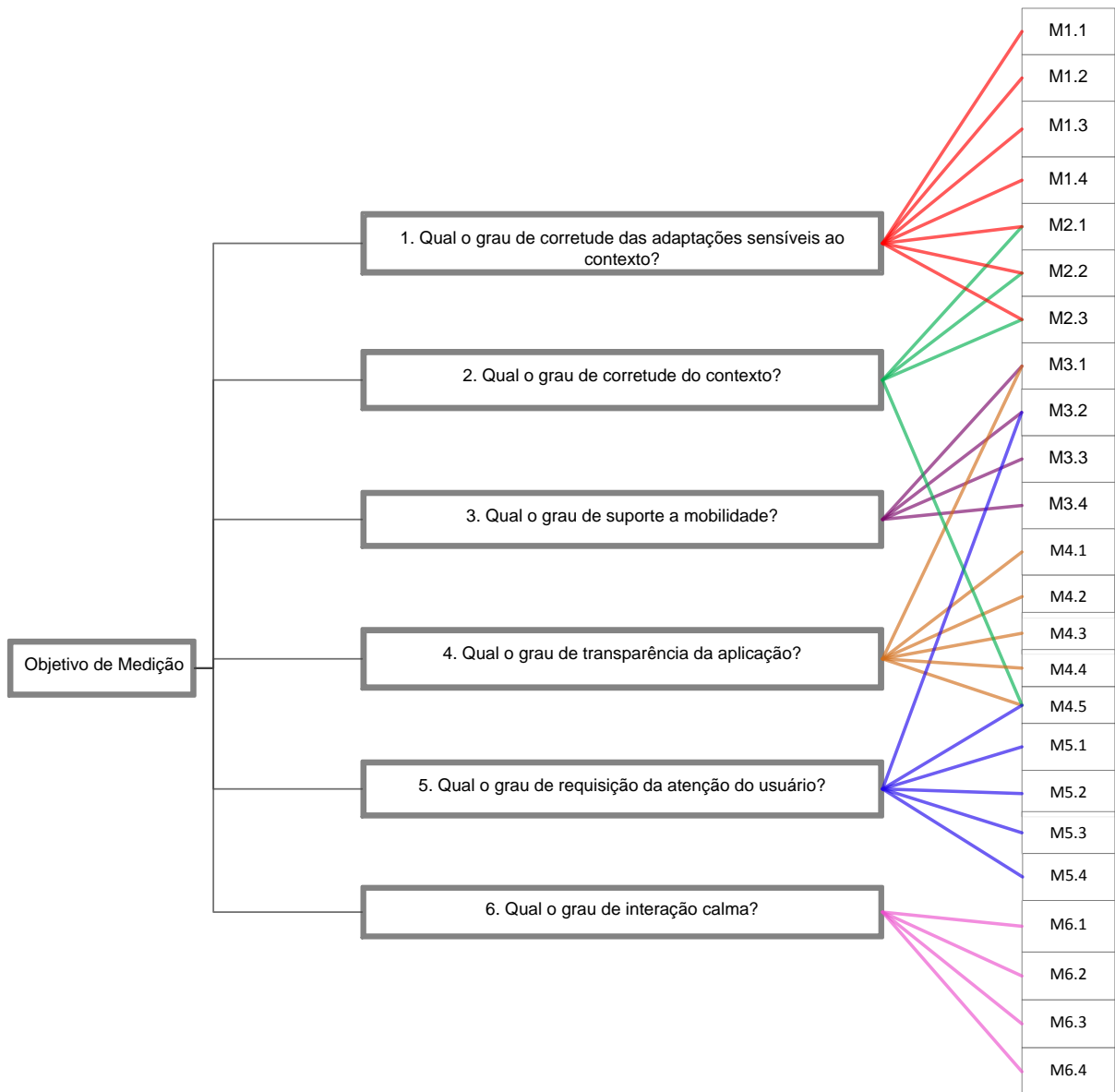
- Alto: a interação ocorre sempre no momento correto. Quando a interação ocorre, o usuário concorda que foi no momento correto e o usuário não quer interagir quando ela não ocorre.

- Média: a interação ocorre na maioria das vezes no momento correto. Poucas vezes acontece do usuário não querer a interação quando ela ocorre, ou de querer interagir quando não ocorre.
- Baixa: Na maioria das vezes, o usuário não concorda com a interação.
- Muito baixa: O contexto não tem efeito sobre quando a interação é oferecida.

Para a medida M6.4, é necessário seguir as seguintes direções:

- Alta: a aplicação suporta ou substitui ações dos usuários. Dependendo da necessidade de cada situação, a aplicação se move da total invisibilidade para a periferia ou centro de atenção. A aplicação se move para o centro da atenção, somente quando absolutamente necessário. A interação é implementada de modo que o usuário julga como natural.
- Média: a aplicação exige desnecessariamente a atenção do usuário, mas a perturbação não é grave. A periferia não é utilizada quando seria suficiente apresentar informações na periferia. O usuário é capaz de continuar suas ações normais.
- Baixa: o usuário precisa mudar o foco de atenção de suas atividades para o serviço ou aplicativo.
- Muito baixo: O usuário precisa interromper sua atividade em andamento para se concentrar completamente na interação com o sistema.

Na Figura 4.7, um resumo com todas as medidas de *software* propostas para responder as questões e, assim, atender ao objetivo de medição é apresentada.



**Figura 4.7** – Resumo dos objetivos de medição, questões e medidas

## 4.4 Coleta de Dados

Essa fase do GQM consiste em coletar, armazenar e processar todos os dados necessários para calcular as medidas definidas. Para isso, os procedimentos de coleta desses dados devem ser planejados. Os procedimentos de coleta definem todos os aspectos necessários para executar a coleta e, para tanto, se preocupam com as seguintes questões: Para uma determinada medida, quem deve coletar os dados necessários? Quando o dado deve ser coletado? Como o dado pode ser coletado mais eficientemente? Para quem deve ser entregue os dados coletados? (VAN SOLINGEN; BERGHOUT, 1999).

Nesse trabalho, para coletar as medidas estabelecidas foi necessário utilizar três diferentes métodos de coleta de dados:

- Geração de *logs* de interação do usuário com a aplicação através da instrumentação do código. Esses logs devem ser coletados durante o uso da aplicação;
- Formulário *on-line* de perguntas para o desenvolvedor da aplicação. Esse deve ser coletado antes do uso da aplicação (apresentado no Apêndice E);
- Formulários *on-line* de perguntas ao usuário. Um deve ser coletado antes da utilização da aplicação e o outro depois do uso da aplicação (apresentados nos Apêndices C e D); e
- Formulário manual de observação para o avaliador deve ser utilizado durante o uso da aplicação pelo usuário (apresentado no Apêndice F).

Em cada definição das medidas na seção anterior, o método de coleta necessário foi especificado na coluna “método de coleta”.

É importante ressaltar que nesta fase também é feito um período de medição experimental com o propósito de verificar se as ferramentas, *logs* e formulários estão prontos para a coleta dos dados com os desenvolvedores e usuários. Essa verificação foi feita durante um dia pelo time GQM desse trabalho.

## 4.5 Interpretação dos Resultados

Nesta fase do GQM, os resultados obtidos na coleta dos dados são utilizados para interpretar as questões e os objetivos de medição definidos. Como apresentado nas tabelas contendo as definições das medidas, os valores de interpretação representam aproximações do que seria um bom resultado e algumas escalas foram definidas específicas para as medidas, isso

porque não existem valores históricos com os quais podemos comparar e especificar uma escala de interpretação mais específica.

## 4.6 Considerações Finais

Neste capítulo foi apresentado um conjunto de medidas de *software* para avaliação de características de IHC em sistemas ubíquos. Esse conjunto foi especificado através da abordagem *Goal-Question-Metric* (GQM), que consiste em reuniões para realizar entrevistas com especialistas no assunto. Essas reuniões também foram guiadas por uma lista de medidas de *software* encontradas pelo MS realizado (apresentada no Apêndice A).

No total foram definidas 24 medidas para avaliar 6 questões de 1 objetivo de medição. O modelo final do GQM gerado é apresentado na Figura 4.7. Essas medidas serão aplicadas em sistemas ubíquos no próximo capítulo, a fim de avaliar o seu uso em aplicações. Também foram definidos os métodos de coleta das medidas e a forma de interpretação para esse trabalho que serão utilizados no Capítulo 5.

# 5 Estudo de Caso

Este capítulo apresenta a aplicação das medidas definidas no capítulo anterior, o que corresponde as fases de coleta e interpretação do GQM. A Seção 5.1 apresenta o planejamento para executar os estudos de caso. A Seção 5.2 apresenta a coleta e interpretação das medidas na aplicação UbiPrinter. A Seção 5.3 apresenta a coleta e interpretação das medidas na aplicação UbiMute. A Seção 5.4 apresenta a coleta e interpretação das medidas na aplicação GREat Tour. A Seção 5.5 apresenta uma discussão geral dos resultados e, por fim, a Seção 5.6 apresenta as considerações finais do capítulo.

## 5.1 Planejamento

A aplicação das medidas foi realizada por meio de três estudos de caso (EC). EC é uma metodologia de pesquisa empírica que investiga um determinado fenômeno em um contexto. EC foi escolhida porque permite a coleta de dados quantitativos e qualitativos e é conduzido em configurações do mundo real, tendo, assim, um alto grau de realismo (RUNESON; HÖST, 2008). No entanto, essa metodologia não fornece um nível de controle como no experimento controlado. Assim, podem existir fatores desconhecidos de variação sobre um resultado.

Para planejar o estudo de caso deste trabalho foram seguidas as atividades de planejamento propostas por (RUNESON; HÖST, 2008): (i) definir o objetivo do estudo de caso; (ii) definir os casos a serem estudados; (iii) definir as questões de pesquisa; (iv) definir os instrumentos de coleta de dados; e (v) definir os procedimentos de análise.

O objetivo dos estudos de caso é derivado pela motivação dessa pesquisa que é avaliar a qualidade da IHC de sistemas ubíquos através das medidas de *software* que foram definidas no Capítulo 4. Os casos analisados correspondem a três aplicações ubíquas desenvolvidas no GREat<sup>5</sup>: GREat Tour, UbiPrinter e UbiMute. Elas serão apresentadas nas próximas seções.

As questões de pesquisa são as mesmas definidas pelo GQM na Seção 4.3.4. Os instrumentos para coleta de dados também já foram definidos para cada medida resultante do GQM (Seção 4.2.5). Por fim, para a análise dos focos de qualidade do objetivo, que neste trabalho, são as características de qualidade definidas no MS, uma análise dos resultados das

---

<sup>5</sup> GREat - <http://www.great.ufc.br/>



medidas para cada questão é realizada. Essas questões correspondem às características de qualidade.

Doze usuários foram convidados para participar da avaliação das três aplicações. Todos os usuários possuem formação acadêmica de nível superior para cima. Sete pessoas possuem ensino superior completo, três pessoas são mestrandos e dois doutorandos. Metade dos usuários não é da área de computação, atuando nas áreas de Secretariado, Administração, Financeiro e Recursos Humanos. A outra metade é formada na área da computação. Todos os usuários possuem experiência com o uso de *smartphones* com telas *touch screen* e a maioria (9 deles) já utilizaram aplicações móveis e sensíveis ao contexto.

Três pessoas que participaram do desenvolvimento das três aplicações também foram convidados para participar da avaliação. Essa parte consistia em responder o questionário do desenvolvedor, necessário para coleta de algumas medidas definidas.

Os dispositivos móveis utilizados para teste em duas aplicações (GREat Tour e UbiPrinter) possui as seguintes especificações:

- Versão do Android 4.1.2, Número do modelo GT – I9300, 832 MB de RAM, 11,35 GB de Memória do sistema, Processador Quad Core 1.4GHz, Câmera 8MP, Leitor de QRCode: Barcode Scanner, versão 4.3.1, 1,09MB.
- Versão do Android 4.0.4, Número do modelo GT – I9300, 779 MB de RAM, Memória do sistema: 11,35 GB, Processador Quad Core 1.4GHz, Câmera 8MP, Leitor de QRCode: Barcode Scanner, versão 4.3.1, 1,09MB.

Os cenários de uso foram específicos de cada aplicação e são apresentados nas próximas seções.

## 5.2 UbiPrinter

A UbiPrinter é uma aplicação desenvolvida na plataforma *Android* e tem como objetivo auxiliar na impressão de documentos ao localizar a impressora mais próxima do usuário. A Figura 5.1 apresenta as telas da aplicação.



**Figura 5.1** – Aplicação UbiPrinter

A aplicação funciona da seguinte forma: (i) o usuário inicializa a aplicação e um ícone para abrir um arquivo é apresentado; (ii) ao clicar no ícone, o arquivo pode ser visualizado e um botão para imprimir esse arquivo pode ser acionado; (iii) ao clicar no botão imprimir, a aplicação coleta as redes *wifi* mais próximas, verifica qual a rede com maior intensidade de sinal, e de acordo com essa informação, verifica qual impressora está no alcance dessa rede; e (iv) a aplicação envia o arquivo para a impressão nessa impressora, informando ao usuário.

Assim, essa aplicação utiliza uma informação de contexto: a força do sinal das redes *wifi*, e executa uma adaptação: impressão de arquivo na impressora mais próxima do usuário de acordo com a *wifi* de maior intensidade de sinal.

A tarefa definida para o usuário executar foi imprimir um arquivo já pré-estabelecido pela aplicação. Os doze usuários foram divididos em quatro grupos para executar a aplicação em diferentes sequências de andares e ambientes do GREat, tal divisão é apresentada na Tabela 5.2.

**Tabela 5.1** – Procedimento para Realização da Avaliação do UbiPrinter

Usuários	Andares e Lugares	
Grupo 1	1º andar / Sala de Seminários	2º andar / Sala de Pesquisa
Grupo 2	2º andar / Sala do Administrativo 2	1º andar / Recepção
Grupo 3	1º andar / Sala de Reuniões	2º andar / Sala dos professores
Grupo 4	2º andar / Lobby - Elevador	1º andar / Lab P&D Tools

### 5.2.1 Coleta e Interpretação dos Resultados

Nesta seção são apresentados os resultados obtidos pela coleta de dados das medidas de *software* aplicadas para avaliar a UbiPrinter. Os resultados encontrados são discutidos para cada questão definida no Capítulo 4. No final de cada questão uma conclusão parcial é descrita com base nos resultados das medidas coletadas.

#### Questão 1 - Qual o grau de corretude da adaptação sensível ao contexto?

Para responder a essa primeira questão as medidas M1.1, M1.2, M1.3 e M1.4 foram coletadas. Os resultados encontrados são apresentados na Tabela 5.2 e discutidos abaixo.

**Tabela 5.2** – Resultados da UbiPrinter para a Questão 1

ID da Medida	Nome da Medida	Resultado
M1.1	Corretude da Adaptação	A = 12
		B = 23
		N = 1
		<b>X = 52%</b>
M1.2	Intervalo de mudanças do contexto	00:01:57
M1.3	Tempo de Adaptação	T = 413 milissegundos
M1.4	Grau de Adaptação	A = 23
		B = 29
		N = 1
		<b>X = 79%</b>

A medida M1.1 obteve resultados muito distantes do apropriado, apenas 52% das adaptações ocorreram de maneira correta do ponto de vista do usuário. Essa medida indica que a aplicação falhou muitas vezes em determinar a impressora mais próxima do usuário.

A medida M1.2 obteve um resultado baixo, o que neste caso é um resultado bom, porque foi em minutos, isto é, demora cerca de um minuto para uma mudança contextual acontecer. Indicando que não há chances de uma adaptação ocorrer antes de uma mudança contextual. Esse resultado já era o esperado nesse tipo de aplicação, pois ela somente coleta o contexto quando o usuário faz uma requisição de impressão.

Na medida M1.3, o tempo de adaptação foi curto (milissegundos). A mensagem de escolha da impressora foi apresentada rapidamente para o usuário. No entanto, o desvio-padrão calculado mostra que os valores variaram bastante ( $D_p = 330$  milissegundos).

A medida M1.4 mostrou que apenas em 79% das vezes as adaptações ocorreram, indicando que em algumas situações a aplicação não realiza as adaptações, mesmo ocorrendo uma mudança contextual. Neste caso, foi possível perceber pelos *logs* e pelo avaliador que as adaptações não ocorreram nos seguintes lugares: Lobby-Elevador (2º andar) e Laboratório de P&D Tools (1º andar), por serem lugares onde não há alcance das redes *wifi* que a aplicação estava levando em consideração.

É interessante notar que mesmo os resultados das medidas M1.2 e M1.3 sejam bons resultados, a aplicação provê um baixo grau de corretude das adaptações satisfatório, pois as adaptações nem sempre ocorreram (79%) e quando aconteceram nem sempre estavam corretas (52%). Esse resultado pode ser causado devido ao grau de corretude do contexto, que será avaliado na próxima questão.

## **Questão 2 - Qual o grau de corretude do contexto?**

Para essa segunda questão, as medidas M2.1, M2.2 e M2.3 foram coletadas. Os resultados encontrados são apresentados na Tabela 5.3 e discutidos a seguir.

**Tabela 5.3 - Resultados da UbiPrinter – Questão 2**

ID da Medida	Nome da Medida	Resultado
M2.1	Corretude do contexto	A = 12
		B = 29
		M = 1
		X = 41%
M2.2	Corretude das situações de contexto	Não aplicável
M2.3	Granularidade do Contexto	Informação de Contexto de Baixo Nível

A medida M2.1 obteve resultado muito distante do apropriado, apenas 41% das informações contextuais coletadas correspondiam à real localização do usuário dentro do GREat.

A medida M2.2 não foi aplicada pois a aplicação não utiliza “situações de contexto”, apenas utiliza a informação do sinal das redes *wifi*, que é uma informação de contexto de baixo nível, o que responde à medida M2.3, indicando que a aplicação é muito vulnerável a mudanças ou incertezas, impactando na corretude da adaptação.

A medida M4.5 (Grau de Intervenção do Usuário) foi definida para transparência, mas esta medida pode ser avaliada aqui por possibilitar verificar resultados que indicam conflitos de contexto. Seu resultado foi 0, indicando que durante a interação não aconteceram intervenções do usuário, no sentido de ter que resolver um conflito de contexto encontrado.

Os resultados dessas medidas indicam que o grau de corretude do contexto está baixo pois a maioria das medidas obtiveram um resultado baixo e esse é o motivo pelo qual o grau de corretude das adaptações dessa aplicação também é baixo.

### Questão 3 - Qual o grau de suporte à mobilidade?

Para essa terceira questão, as medidas M3.1, M3.2 e M3.4 foram coletadas. Os resultados encontrados e a interpretação desses resultados são apresentados na Tabela 5.4 e discutidos a seguir.

**Tabela 5.4 - Resultados da UbiPrinter – Questão 3**

ID da Medida	Nome da Medida	Resultado
M3.1	Indicador de Mobilidade Transparente	Inexistente
M3.2	Grau de funcionamento da rede	X = 33
M3.4	Suporte ao acesso do sistema	X = 8

A medida M3.1 obteve resultado “inexistente”, pois a aplicação não realiza nenhum tipo de mobilidade, nem mesmo guarda a sessão do usuário, uma vez que não possui acesso através de login. Além disso, ela é não é capaz, caso seja instalada em outro dispositivo, de se adaptar aos novos recursos, como por exemplo, o tamanho de tela.

A medida M3.2 apresenta um resultado muito longe de zero. Isso aconteceu, principalmente, porque a aplicação não carregou o arquivo para imprimir, pois estava localizado na internet e não localmente.

A medida M3.4 obteve um bom resultado, o dispositivo utilizado para uso das aplicações possui 8 tecnologias de acesso à comunicação (2G; 2,5G; 2,75G; 3G; H, H+, *Wifi*, *Bluetooth*), permitindo assim o acesso a uma tecnologia, caso a que está sendo utilizada falhe.

Por fim, os resultados dessas medidas indicam que o grau de suporte à mobilidade também está baixo, pois a maioria das medidas (M3.1 e M3.2) não obtiveram bons resultados.

#### Questão 4 - Qual o grau de transparência da aplicação?

Para essa quarta questão, as medidas M4.1, M4.2, M4.3, M4.4 e M4.5 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.5 e discutidos a seguir.

Tabela 5.5

<b>Tabela 5.5 - Resultados UbiPrinter – Questão 4</b>		
<b>ID da Medida</b>	<b>Nome da Medida</b>	<b>Resultado</b>
M4.1	Grau de interação implícita	A = 1
		N = 4
		$X = 0,25 * 100 = 25\%$
M4.2	Tipo de comportamento adaptativo	Comportamento adaptativo transparente
M4.3	Proatividade da aplicação	N = 2
		A = 2
		X = 0
M4.4	Grau de esforço para inicializar o sistema	A = 1
M4.5	Grau de intervenção do usuário	0

O elemento de medição N da medida M4.1 é quatro, pois a aplicação possui as seguintes entradas explícitas: 1) abrir o arquivo, 2) abrir menu, e 3) requisitar impressão. Como entrada implícita apresenta: 1) escolha da impressora que o documento será impresso, resultando assim em N = 4. O resultado dessa medida foi razoável pois apresentou 25% de interação implícita. A

aplicação poderia substituir ações como: a requisição de impressão, considerando o fato de que se o usuário está abrindo o arquivo nesse tipo de aplicação é porque ele já deseja que seja impresso o arquivo.

A medida M4.2, coletada a partir do desenvolvedor, indica que o tipo de comportamento é adaptativo transparente. No entanto, após o uso da aplicação, sete pessoas conseguiram observar que a localização estava sendo determinada pelas redes *wifi*. Vale ressaltar que seis dessas pessoas eram da computação e possuem conhecimentos sobre aplicações móveis e sensíveis ao contexto. Somente uma pessoa que não era da computação conseguiu perceber, assim o resultado pode ser considerado como adaptativo transparente.

Na medida M4.3, o elemento de medição N é 2, pois o número total de ações desenvolvidas que podem ser apoiadas pelos sensores é dois: 1) escolher uma impressora e 2) confirmar impressão. Com a utilização do sensor (algoritmo que captura a força do sinal de redes *wifi*), essas duas ações são substituídas (escolher uma impressora e confirmar impressão). Sendo assim, essa medida apresenta um bom resultado.

A medida M4.4 considerou que o único grau de esforço para inicializar a aplicação é a habilitação do sensor *wifi*, no entanto, essa habilitação normalmente é feita pelos usuários em seus celulares, então não representa um grande esforço. Ainda assim, esse ponto poderia ser melhorado pela própria aplicação ao ser implementada uma função de habilitação desse sensor.

A medida M4.5 obteve resultado 0, nenhum esforço foi necessário para que a adaptação ocorresse, o que representa um resultado bom. Tal resultado foi obtido porque em nenhum momento a aplicação requisitou que o usuário confirmasse sua localização, no entanto, se o resultado dessa medida tivesse sido maior, provavelmente, a corretude do contexto seria maior também, no entanto, a transparência da aplicação seria afetada.

Outras medida que pode ser analisada para apoiar a interpretação dessa questão é a M3.1 (Indicador de Mobilidade Transparente), que apresentou resultado “inexistente”.

Por fim, a aplicação obteve um grau de transparência da aplicação alto, pois a maioria das medidas definidas para essa questão (M4.1, M4.2, M4.3, M.4) tiveram bons resultados.

### **Questão 5 - Qual o grau de requisição da atenção do usuário?**

Para essa quinta questão, as medidas M5.1, M5.2, M5.3 e M5.4 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.6 e discutidos a seguir.

**Tabela 5.6 - Resultados UbiPrinter – Questão 5**

ID da Medida	Nome da Medida	Resultado
M5.1	Quantidade de mudanças de foco	A = 15
M5.2	Grau reconhecido de tempo perdido durante o uso	$\sum$ Tempo de Adaptação = 0,15 min $\sum$ Tempo para Carregar Arquivo = 4,5 min X = 4,65 minutos
M5.3	Número de entradas explícitas	3
M5.4	Tempo médio entre falhas	T = 58 minutos A = M3.2 = 33 X = 1,8

A medida M5.1 indica a frequência com que o usuário precisou mudar de foco para a tecnologia. Durante o uso, 15 vezes os usuários mudaram de foco. Essa mudança correspondeu às seguintes ações: reiniciar o sensor *wifi* (6), trocar de conexão (2), reiniciar a aplicação (6), e reiniciar o celular (1). Tudo isso ocorreu porque o arquivo demorou muito para ser carregado e a primeira reação dos usuários é fazer alguma ação para que o arquivo seja carregado.

A medida M5.2 busca identificar os tempos considerados “perdidos”. Esse tempo perdido é o tempo que o usuário gasta esperando o retorno da aplicação. Para calcular o elemento de medição X (Somatório dos tempos perdidos), foram contabilizados os seguintes tempos: o somatório dos tempos de adaptação e o somatório dos tempos para carregar o arquivo. O resultado indica que o tempo para carregar o arquivo foi alto, mas o tempo de adaptação foi baixo. No entanto, não foi contabilizado nessa medida o tempo que o usuário gastou trocando de foco, por exemplo, ligando ou reiniciando um sensor para que a aplicação funcionasse. Essa contabilização poderia levar essa medida a ter um resultado muito pior.

A medida M5.3 indica que o número de entradas explícitas (do usuário) está baixo. A medida M5.4 representa o tempo médio entre as falhas ocorridas. Neste caso somente ocorreram falhas de conexão, o que é calculado através da medida M3.4. O resultado, então, foi 1,8 minutos, um resultado próximo do número zero, o que não representa um bom resultado.

Assim, pode-se concluir, que o grau de requisição da atenção do usuário nessa aplicação não apresenta um bom resultado pois a maioria das medidas apresentam resultados ruins.



## Questão 6 - Qual o grau de calma da interação?

Para essa sexta e última questão, as medidas M6.1, M6.2, M6.3 e M6.4 foram utilizadas. Essas medidas foram respondidas pelo próprio usuário da aplicação e os resultados são apresentados na Tabela 5.7.

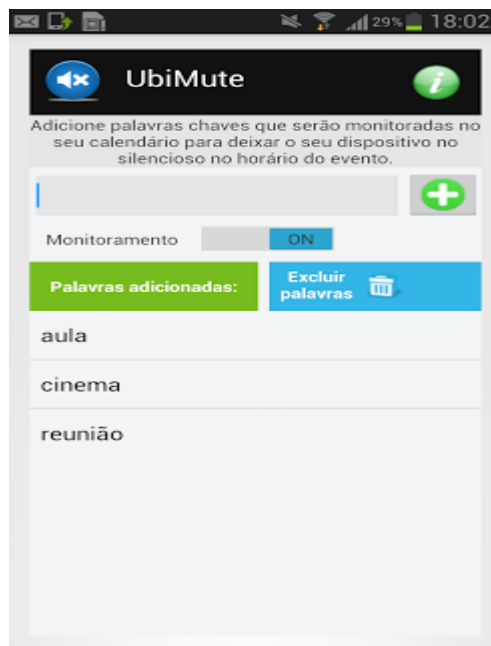
**Tabela 5.7 - Resultados UbiPrinter – Questão 6**

ID da Medida	Nome da Medida	Resultado
M6.1	Grau de disponibilidade	Alto = 1 <b>Médio = 7</b> Baixo = 2 Muito Baixo = 2
M6.2	Grau de relevância da interação	<b>Alto = 7</b> Médio = 4 Baixo = 1 Muito Baixo = 0
M6.3	Grau de sincronia da interação	Alto = 3 <b>Médio = 5</b> Baixo = 3 Muito Baixo = 1
M6.4	Grau de cortesia da interação	<b>Alto = 6</b> Médio = 4 Baixo = 2 Muito Baixo = 0

Os resultados em geral foram bons, nenhuma medida resultou em “muito baixo”. Somente as medidas M6.1 e M6.3 obtiveram resultado “médio”. Isso provavelmente aconteceu devido às adaptações não terem ocorrido para metade dos usuários e porque o arquivo para impressão não foi carregado no momento que os usuários precisavam.

## 5.3 UbiMute

A UbiMute é uma aplicação desenvolvida na plataforma Android, sendo um serviço que executa em *background* no dispositivo móvel monitorando o *Google Calendar* do usuário em busca de eventos nos quais o usuário não pode receber ligações, como “reunião”, “aula” ou “cinema”. Ao localizar um evento desses, a aplicação coloca o dispositivo móvel do usuário no silencioso, a fim de que o usuário não receba ligações durante esses eventos. A Figura 5.2 apresenta o seu funcionamento.



**Figura 5.2** – Aplicação UbiMute

A UbiMute permite que o usuário especifique quais eventos ele gostaria que a aplicação colocasse no silencioso, após isso o usuário pode configurar a aplicação para executar o monitoramento dos eventos.

Para executar o estudo de caso com essa aplicação, foi necessário recrutar usuários que já utilizassem o *Google Calendar* (agenda do *Google*) para agendar eventos como “reuniões”, “aulas”, entre outros que exigissem o celular no silencioso. Outro requisito é que os usuários deveriam possuir o celular na plataforma Android, na qual a aplicação foi desenvolvida. Assim, o usuário poderia vivenciar o uso da aplicação em um contexto real.

A tarefa definida para o usuário executar foi cadastrar pelo menos um evento que iria acontecer ao longo da semana no calendário, bem como uma palavra chave corresponde ao nome desse evento na aplicação UbiMute, para que ela iniciasse o monitoramento desse evento.

Dos doze usuários voluntários, somente oito conseguiram executar a aplicação em seus *smartphones*, nos outros 4, a aplicação não executava nos *smartphones*. Portanto, foram considerados apenas os dados dos oito usuários.

### **5.3.1 Coleta e Interpretação dos Resultados**

Nesta seção são apresentados os resultados obtidos pela coleta de dados na aplicação UbiMute. Cada questão é comentada de acordo com os resultados das medidas obtidas.

### Questão 1 - Qual o grau de corretude da adaptação sensível ao contexto?

Para responder a essa primeira questão as medidas M1.1, M1.2, M1.3 e M1.4 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.8 e discutidos a seguir.

**Tabela 5.8 - Resultados da UbiMute para a Questão 1**

ID da Medida	Nome da Medida	Resultado
M1.1	Corretude da Adaptação	A = 6 B = 22 N = 1 X = 27%
M1.2	Intervalo de mudanças do contexto	01:15:12
M1.3	Tempo de Adaptação	T = 63 milissegundos
M1.4	Grau de Adaptação	A = 22 B = 24 N = 1 X = 92%

A medida M1.1 obteve resultados muito distante do apropriado, apenas 27% das adaptações ocorreram de maneira que corresponde ao contexto atual do usuário.

A medida M1.2 obteve um resultado baixo, porque foi em horas, isto é, demora cerca de uma hora para uma mudança contextual acontecer, o que indica, assim como na aplicação UbiPrinter, que não é possível uma adaptação ocorrer antes de uma mudança contextual. Esse resultado também já era o esperado nesse tipo de aplicação, pois a aplicação leva em consideração “eventos dos usuários” cadastrados na agenda e um evento normalmente leva mais que minutos para encerrar.

Na medida M1.3, o tempo de adaptação foi muito curto (66 milissegundos). Esse tempo já era esperado porque a aplicação coloca o celular no silencioso no horário exato que o evento foi cadastrado na agenda do celular. (Dp = 112 milissegundos)

A medida M1.4 mostrou um alto grau de adaptação (92%), indicando que a aplicação se adaptou na grande maioria das vezes em que uma mudança contextual foi coletada.

Apesar de três medidas terem um bom resultado (M1.2, M1.3 e M1.4), a corretude das adaptações não foi considerada aceitável, porque a M1.1 foi muito baixa. Da mesma forma que aconteceu na UbiPrinter, esse resultado pode ter sido causado devido a corretude do contexto, que é apresentada a seguir.

## Questão 2 - Qual o grau de corretude do contexto?

Para essa segunda questão, as medidas M2.1, M2.2, M2.3 e M2.4 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.9 e discutidos a seguir.

**Tabela 5.9 - Resultados da UbiMute para a Questão 2**

ID da Medida	Nome da Medida	Resultado
M2.1	Corretude do contexto	A = 2 B = 22 N = 1 X = 27%
M2.2	Corretude das situações de contexto	Não aplicável
M2.3	Granularidade do Contexto	Informação de contexto de baixo nível

A medida M2.1 obteve resultados muito distantes do apropriado, apenas 27% das informações de contexto coletadas correspondiam à real atividade do usuário. Já a medida M2.2 não foi aplicada pois a aplicação não utiliza “situações de contexto”, então não é possível calcular a corretude dessas situações. Isso já responde a medida M2.3, isto é, a aplicação utiliza somente informação de contexto de baixo nível. Por fim, a medida M2.4 obteve um resultado baixo, pois a aplicação utiliza somente uma informação (agenda) para determinar a atividade do usuário.

Essas medidas indicam que a corretude do contexto coletado foi muito baixa. Assim, é possível observar que se a aplicação utilizasse outras informações contextuais para inferir a atividade do usuário, as medidas M1.1 e M2.1 poderiam ter resultados muito diferentes. O que aconteceu é que mesmo as pessoas cadastrando eventos que realmente iriam acontecer durante a semana, imprevistos ocorreram e essas pessoas não puderam comparecer no evento e se esqueceram de retirar da agenda o evento cadastrado. Isso resultou em algumas situações desconfortáveis para o usuário, como por exemplo: um usuário, que não compareceu ao evento cadastrado na agenda, deixou de atender uma ligação importante porque o celular estava no silencioso. Outro usuário, que também não foi ao evento cadastrado, recebeu uma ligação durante esse período em que o celular estava no silencioso e somente percebeu a tempo de atender a ligação por coincidência. Essas informações foram coletadas através do formulário no Apêndice D.

É necessário que outras informações de contexto sejam levadas em consideração para que a aplicação realize as adaptações de forma mais confiável. Um exemplo de informação de

contexto que poderia ser adicionada nessa aplicação é a localização do usuário. Assim, mesmo que a agenda indique um compromisso, se o usuário não estivesse no local do evento, a aplicação não colocaria o celular do usuário no silencioso.

### Questão 3 - Qual o grau de suporte a mobilidade?

Para essa terceira questão, as medidas M3.1, M3.3 e M3.4 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.10 e discutidos a seguir.

**Tabela 5.10 - Resultados da UbiMute para a Questão 3**

<b>ID da Medida</b>	<b>Nome da Medida</b>	<b>Resultado</b>
M3.1	Indicador de Mobilidade Transparente	Inexistente
M3.2	Grau de funcionamento da rede	0
M3.4	Suporte ao acesso do sistema	2

A medida M3.1 indica que a aplicação não consegue realizar nenhum tipo de mobilidade. A medida M3.2 foi coletada através de questionários com os usuários após o uso e seu resultado foi zero. Isso porque essa aplicação depende muito pouco de conexão com a rede. A única dependência existente é que caso o usuário cadastre um evento na agenda através do navegador em um *desktop*, é necessário que aconteça uma sincronização com a agenda local do usuário no celular, através de uma conexão com a rede. Portanto, a única falha que pode ocorrer é, por falta de conexão, o celular não conseguir sincronizar os eventos cadastrados e assim a aplicação não conhecer esse evento. No entanto, esse cenário é muito difícil de ocorrer, pois a UbiMute realiza um monitoramento de trinta em trinta segundos.

A medida M3.4 foi parcialmente coletada, pois a aplicação foi utilizada nos dispositivos móveis dos usuários, somente foi possível identificar que os celulares possuíam 3G e *Wifi*.

Por fim, apesar de não apresentar falhas de conexão, a aplicação não possui uma mobilidade transparente, característica muito importante na computação ubíqua, portanto, é preciso melhorar o grau de suporte à mobilidade.

### Questão 4 - Qual o grau de transparência da aplicação?

Para essa quarta questão, as medidas M4.1, M4.2, M4.3, M4.4 e M4.5 foram coletadas. Os resultados encontrados são apresentados na Tabela 5.11 e discutidos a seguir.

**Tabela 5.11 - Resultados da UbiMute para a Questão 4**

<b>ID da Medida</b>	<b>Nome da Medida</b>	<b>Resultado</b>
M4.1	Grau de interação implícita	A = 1
		N = 3
		$X = 0,333 * 100 = 33,3\%$
M4.2	Tipo de comportamento adaptativo	Comportamento adaptativo não transparente
M4.3	Proatividade da aplicação	N = 2
		A = 2
		X = 0
M4.4	Grau de esforço para inicializar o sistema	A = 2
M4.5	Grau de intervenção do usuário	0

Para calcular a medida M4.1, foram consideradas as seguintes entradas explícitas: adicionar palavras chave e ativar o monitoramento, e como entrada implícita: eventos da agenda do usuário. O resultado dessa medida pode ser considerado bom, dentre três entradas, uma é implícita.

A medida M4.2, coletada a partir do desenvolvedor, indica que o tipo de comportamento é adaptativo não transparente, pois o usuário sabe que a aplicação está monitorando sua agenda. Caso a aplicação utilize outras informações de contexto, o resultado dessa medida pode ser diferente.

A medida M4.3 foi calculada da seguinte forma: número de ações que podem ser substituídas por sensores: duas (colocar e retirar o *smartphone* do usuário do silencioso), que também são as ações que a aplicação pode substituir. Assim, essa medida apresentou um resultado muito bom.

A medida M4.4 considerou que o grau de esforço para inicializar a aplicação são as duas entradas que ela requisita para o usuário: adicionar palavras chave e iniciar o monitoramento. Vale ressaltar neste resultado, que essas duas ações são necessárias apenas uma vez, pois uma vez realizadas, dificilmente o usuário precisa voltar para realizá-las novamente.

A medida M4.5 obteve resultado 0, nenhum esforço foi necessário para que a adaptação ocorresse. Esse resultado foi obtido porque em nenhum momento a aplicação requisitou que o

usuário confirmasse se poderia ou não colocar o celular no silencioso, no entanto, a corretude da adaptação não foi alta. Provavelmente, se o resultado dessa medida tivesse sido maior, a corretude da adaptação também seria.

Outras medida que pode ser avaliada nesta questão é a M3.1 (Indicador de Mobilidade Transparente), que apresentou resultado “inexistente”.

Por fim, a aplicação obteve um grau de transparência da aplicação alto, pois a maioria das medidas definidas para essa questão (M4.1, M4.2, M4.3, M.4) tiveram bons resultados.

### Questão 5 - Qual o grau de requisição da atenção do usuário?

Para essa quinta questão, as medidas M5.1, M5.2, M5.3 e M5.4 foram utilizadas. Os resultados encontrados são apresentados na Tabela 5.12 e discutidos a seguir.

**Tabela 5.12 - Resultados UbiMute– Questão 5**

ID da Medida	Nome da Medida	Resultado
M5.1	Quantidade de mudanças de foco	0
M5.2	Grau reconhecido de tempo perdido durante o uso	$\sum$ Tempo de Adaptações = 63 milissegundos
M5.3	Número de entradas explícitas	2
M5.4	Tempo médio entre falhas	$T =$ aproximadamente 14 dias $A = 0$

A medida M5.1 indica a frequência com que o usuário precisa mudar de foco para a tecnologia. Durante os 14 dias de uso da aplicação, os usuários não tiveram que mudar o foco desnecessariamente para essa medida, portanto, essa medida não foi calculada.

A medida M5.2 busca identificar o tempo perdido pelo usuário utilizando a aplicação. Para calcular o elemento de medição X (Somatório dos tempos perdidos), foi contabilizado somente o somatório dos tempos de adaptação (63 milissegundos). O resultado indica que o tempo perdido foi muito baixo. No entanto, não foi contabilizado nessa medida o tempo em que o usuário gastou adicionando a palavra e iniciando o monitoramento, mesmo assim, esse cálculo não levaria a um grau de tempo de uso maior.

A medida M5.3 indica que o número de entradas explícitas (do usuário) está baixo, assim, a aplicação não requisita muito a atenção do usuário. Vale ressaltar que uma vez adicionando as palavras chave nessa aplicação, dificilmente o usuário precisaria voltar a adicionar novamente. Essa não é uma ação que o usuário deve fazer sempre para utilizar a aplicação.

A medida 5.4 também não foi calculada, pois não aconteceram falhas na aplicação. Assim, pode-se concluir que essa aplicação requisita muito pouco a atenção do usuário.

### Questão 6 - Qual o grau de calma da interação?

Para essa sexta e última questão, as medidas M6.1, M6.2, M6.3 e M6.4 foram coletadas. Os resultados são descritos Tabela 5.13 e discutidos a seguir.

**Tabela 5.13 - Resultados UbiMute – Questão 6**

<b>ID da Medida</b>	<b>Nome da Medida</b>	<b>Resultado</b>
M6.1	Grau de disponibilidade	<b>Alto = 7</b>
		Médio = 0
		Baixo = 0
		Muito Baixo = 1
M6.2	Grau de relevância da interação	<b>Alto = 6</b>
		Médio = 2
		Baixo = 0
		Muito Baixo = 0
M6.3	Grau de sincronia da interação	Alto = 3
		Médio = 0
		Baixo = 0
		<b>Muito Baixo = 5</b>
M6.4	Grau de cortesia da interação	<b>Alto = 6</b>
		Médio = 1
		Baixo = 0
		Muito Baixo = 1

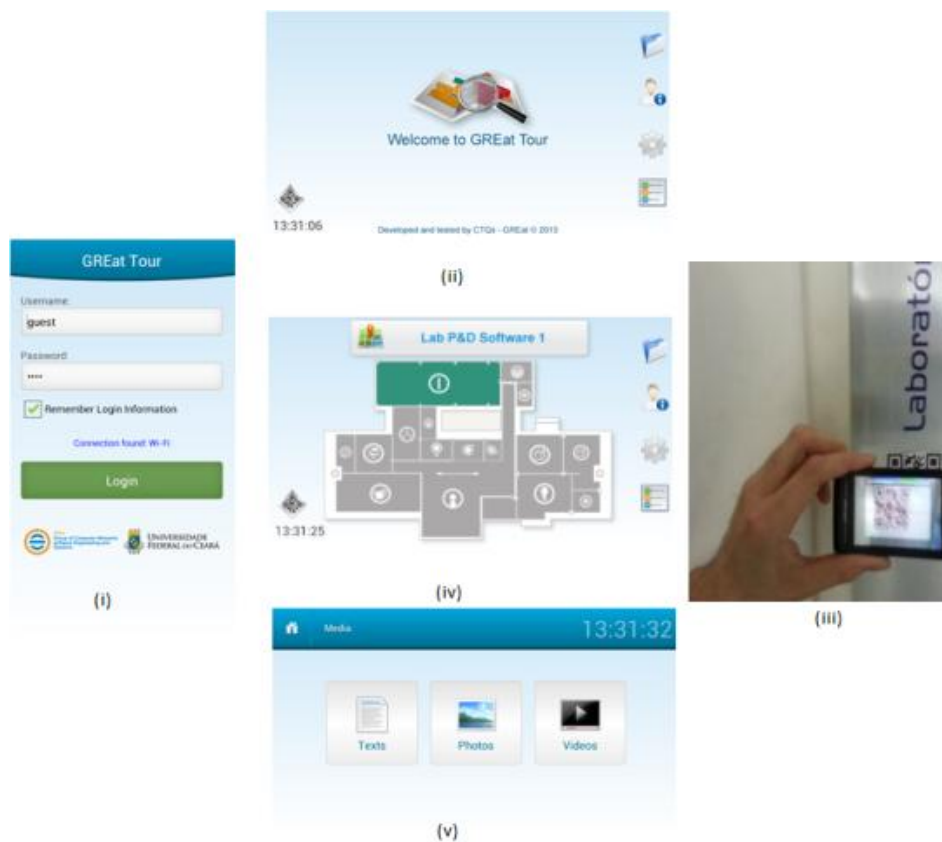
Os resultados em geral foram bons para as medidas M6.1, M6.2 e M6.4. A única medida que obteve resultado muito baixo foi a M6.3, que é o grau de sincronia da interação. Isso aconteceu exatamente porque muitas das adaptações não ocorreram de acordo com o contexto real do usuário, por exemplo, sua localização não foi levada em consideração para realizar adaptações com um grau de correteude maior.

## 5.4 GREat Tour

O GREat Tour é uma aplicação gerada de uma Linha de Produtos de Software Móvel e Sensível ao Contexto (MARINHO et al., 2012). Essa aplicação foi desenvolvida na plataforma Android e se propõe a ser um guia de visitas do laboratório GREat. Ela executa no dispositivo móvel do visitante e provê informações (textos, imagens e vídeos) sobre os ambientes do



laboratório que o usuário está visitando (LIMA et al., 2013). A Figura 5.1 apresenta algumas telas da aplicação.



**Figura 5.3** - Aplicação GREat Tour, adaptada de (LIMA et al., 2013)

A aplicação funciona da seguinte forma: (i) o usuário recebe um *login* e uma senha para sua autenticação, (ii) é exibida uma tela de boas-vindas, na qual o usuário tem a opção de atualizar a localização, (iii) o usuário deve apontar o dispositivo para um QR Code implantado na porta do ambiente para se localizar, (iv) o mapa do laboratório é exibido, destacando-se o ambiente em que o usuário se encontra, e (v) o usuário pode visualizar as seguintes opções de mídia: textos, fotos e vídeos sobre o ambiente que está visitando, porém, a visualização da mídia dependerá do nível de bateria do dispositivo. Quando a bateria está baixa (0-9%), somente texto é exibido, quando ela está média (10-20%), textos e imagens são exibidos e, por fim, quando está alta (21-100%), textos, imagens e vídeos são exibidos.

Assim, essa aplicação apresenta dois tipos de adaptações ( $N=2$ ) e dois tipos de informações de contexto ( $M=2$ ). A primeira adaptação é a visualização do mapa de acordo com a localização do usuário. A segunda adaptação corresponde à exibição do tipo de mídia de acordo com o nível de bateria.

Neste estudo de caso, somente seis dos doze usuários utilizaram a aplicação, devido a indisponibilidade dos outros usuários. Cada usuário fez três visitas (*tours*) ao laboratório do GREat e cada visita consistia em visualizar três ambientes. Em cada ambiente, os usuários atualizavam o mapa e visualizavam todas as mídias disponíveis. O procedimento para realização dos testes é apresentado na Tabela 5.14.

**Tabela 5.14 - Procedimento para Realização da Avaliação do GREat Tour**

		Grupo 1	Grupo 2	Grupo 3
Tour	Ambientes Visitados	Nível de Bateria		
1	Sala de Seminários	Alta	Baixa	Média
	Biblioteca			
	Administrativo 1			
2	Sala de Prototipação	Média	Alta	Baixa
	Laboratório de P&D 1			
	Sala de Reuniões			
3	Copa	Baixa	Média	Alta
	Administrativo 2			
	Laboratório de Pesquisa			

Como apresentado na Tabela 5.15, cada visita (*tour*) foi feita com um dispositivo em diferente nível de bateria, isso com o intuito do usuário vivenciar a adaptação baseada no nível de bateria. Assim, os usuários foram divididos em três grupos de duas pessoas para executar o teste em diferentes sequências de nível de bateria.

### 5.4.1 Coleta e Interpretação dos Resultados

Nesta seção são apresentados os resultados obtidos pela coleta de dados para calcular as medidas de software do capítulo anterior. Também são discutidas as questões da Seção 4.2.4.

#### Questão 1 - Qual o grau de corretude da adaptação sensível ao contexto?

Para essa questão foram coletadas as medidas M1.1, M1.2, M1.3 e M1.4. Os resultados encontrados são apresentados na Tabela 5.16 e discutidos a seguir.

**Tabela 5.15** – Resultados GREat Tour – Questão 1

ID da Medida	Nome da Medida	Resultado
M1.1	Corretude da Adaptação	100%
M1.2	Intervalo de mudanças do contexto	00:01:03
M1.3	Tempo de Adaptação	T = 18332 milissegundos
M1.4	Grau de Adaptação	72%

As medidas M1.1 e M1.2 obtiveram bons resultados. Na M1.1, todas as adaptações que ocorreram durante o uso da aplicação estavam corretas. Na medida M1.2, a média do intervalo de contexto foi alto pois a média do intervalo de mudança do contexto foi em minutos.

A M1.4 não obteve um resultado tão bom quanto a M1.1 pois a aplicação conseguiu se adaptar em 72% das vezes em que uma mudança contextual ocorria, no entanto, esse valor é considerado acima da média. A M1.3 foi considerada ruim pois em média o tempo de adaptação foi 18 segundos. Isso foi posteriormente investigado e foi visto que em alguns ambientes da visita o sinal da rede fica muito baixo e o dispositivo móvel não trocava de rede rapidamente e assim o novo mapa demorava muito para aparecer.

Como a maioria dos resultados foram bons (M1.1, M1.2, M1.4), pode-se concluir que o grau de adaptação sensível ao contexto é alto. A única melhoria que deve ser feita é no tempo de adaptação (M1.3).

### Questão 2 - Qual o grau de corretude do contexto?

Para essa questão foram coletadas as medidas M2.1, M2.3 e M2.4. Os resultados encontrados são apresentados na Tabela 5.16 e discutidos a seguir.

**Tabela 5.16** – Resultados GREat Tour – Questão 2

ID da Medida	Nome da Medida	Resultado
M2.1	Corretude do contexto	100%
M2.2	Corretude das situações de contexto	Não aplicável
M2.3	Granularidade do Contexto	Informação de Contexto de Baixo Nível

A medida M2.2 não foi coletada pelos mesmos motivos que as aplicações UbiPrinter e UbiMute não a coletaram. A medida M2.1 indica que todos os contextos coletados estavam corretos. A medida M2.3 indica que a aplicação utiliza informação de contexto de baixo nível, visto que a aplicação utiliza somente o QRCode e o nível de bateria. No entanto, apesar do resultado baixo, essas informações contextuais foram suficientes para determinar corretamente o

contexto requisitado. Assim, o grau de corretude do contexto da aplicação GREat Tour pode ser considerado alto.

### Questão 3 - Qual o grau de suporte a mobilidade?

Para essa questão foram coletadas as medidas M3.1, M3.2 e M3.4. Os resultados encontrados são apresentados na Tabela 5.17 e discutidos a seguir.

**Tabela 5.17** – Resultados GREat Tour – Questão 3

ID da Medida	Nome da Medida	Resultado
M3.1	Indicador de Mobilidade Transparente	Inexistente
M3.2	Grau de Funcionamento da Rede	X = 6
M3.4	Suporte ao acesso do sistema	X = 8

A medida M3.1 indica que o GREat Tour não possui nenhum tipo de mobilidade e no caso dessa aplicação, a mobilidade é um fator muito importante e essa medida deve ter um peso maior sobre as outras, pois caso a bateria do dispositivo do visitante venha a falhar durante o *tour*, a mobilidade permitiria que o usuário trocasse de dispositivo e continuasse o *tour* a partir do ambiente em que parou.

A medida M3.2 indica que algumas falhas de conexão ocorreram durante o uso. As falhas contabilizadas foram aquelas observadas pelo avaliador e pelo usuário durante o uso. Outras falhas, durante a interação ocorreram, mas não é possível saber se foi por causa da conexão com a rede. Portanto, essa medida teria um resultado maior se uma coleta através de logs de conexão com a rede tivesse sido implementado.

A medida M3.4 indica que há 8 tecnologias para acesso à rede (2G; 2,5G; 2,75G; 3G; H, H+, *Wifi*, *Bluetooth*). Isso significa que há um bom suporte para acesso a rede.

Por fim, é possível concluir que o GREat Tour possui um suporte à mobilidade abaixo do apropriado, principalmente por ser um tipo de aplicação que o usuário pode trocar várias vezes de dispositivos móveis durante o uso.

### Questão 4 - Qual o grau de transparência da aplicação?

Para essa quarta questão, as medidas M4.1, M4.2, M4.3, M4.4 e M4.5 foram coletadas. Os resultados encontrados são apresentados na Tabela 5.18 e discutidos a seguir.

**Tabela 5.18** – Resultados GREat Tour – Questão 4

ID da Medida	Nome da Medida	Resultado
		A = 1
M4.1	Grau de interação implícita	N = 17
		$X = 0,058824 * 100 = 5,8824\%$
M4.2	Tipo de comportamento adaptativo	Comportamento adaptativo não transparente
		N = 1
M4.3	Proatividade do sistema	A = 0
		X = 1
M4.4	Grau de esforço para inicializar o sistema	1 (login)
M4.5	Grau de intervenção do usuário	50%

O elemento de medição N da medida M4.1 é dezessete pois a aplicação possui as seguintes entradas explícitas: 1) digitar login, 2) digitar senha, 3) clicar entrar, 4) clicar em mídia, 5) clicar em texto, 6) clicar em imagem\*, 7) clicar em vídeo\*, 8) clicar em texto específico, 9) clicar em imagem específica\*, 10) clicar em vídeo específico\*, 11) acionar QR Code, 12) clicar no ícone pessoas, 13) clicar no ícone de uma pessoa específica, 14) clicar em configurações, 15) clicar em *checkbox* de habilitação da bateria e 16) clicar em legenda. Isso resulta, sendo assim, em dezesseis entradas explícitas. Como entrada implícita, a aplicação GREat Tour possui apenas uma que é: o nível de bateria do dispositivo. Assim, o elemento de medição N é igual a dezessete. O resultado dessa medida foi pouco próximo de 1. A aplicação poderia substituir a ação em que o usuário precisa acionar o QRCode para descobrir sua localização dentro do prédio do GREat.

A medida M4.2, coletada a partir do desenvolvedor, indica que o tipo de comportamento é adaptativo e não transparente, pois o usuário conhece todas as entradas da aplicação.

Na medida M4.3, o elemento de medição N é 1 (obter a localização dentro do prédio do GREat). Essa ação não é substituída por nenhum sensor, pois o usuário é requisitado a interagir com o QRCode para obter a localização *indoor*. Essa medidas poderia ter apresentando um bom resultado se técnicas de localização *indoor* tivessem sido implementadas. Por exemplo, a fusão de sensores magnetômetro, giroscópio e acelerômetro é uma técnica para descobrir a localização *indoor* sem interferência do usuário.

A medida M4.4 considerou que o único grau de esforço para inicializar a aplicação é a ação do usuário realizar o login para poder utilizar a aplicação.

A medida M4.5 obteve resultado 50%. No caso da adaptação da localização no mapa, sempre que o usuário visitava um novo ambiente era necessário apontar o celular para o QRCode. Na adaptação das mídias, nenhuma interferência aconteceu, pois a aplicação não perguntava ao usuário se poderia mesmo não exibir as mídias (vídeos ou imagens) quando a bateria alcançava o um determinado nível de bateria.

Além dessas medidas, a M3.1 (Indicador de Mobilidade Transparente) apresentou resultado “inexistente”.

Assim, como a maioria das medidas (M4.1, M4.2, M4.3, M4.5 e M3.1) apresentou resultados abaixo do apropriado, o grau de transparência da aplicação GREat Tour pode ser considerada baixo.

### Questão 5 - Qual o grau de requisição da atenção do usuário?

Para essa quinta questão, as medidas M5.1, M5.2, M5.3 e M5.4 foram coletadas. Os resultados encontrados são apresentados na Tabela 5.19 e discutidos a seguir.

**Tabela 5.19 - Resultados GREat Tour–Questão 5**

ID da Medida	Nome da Medida	Resultado
M5.1	Quantidade de mudanças de foco	A = 14 ações
M5.2	Grau reconhecido de tempo perdido durante o uso	$\sum$ Adaptações = 1,8 min $\sum$ logins = 3 min $\sum$ texto = 0,04905 min $\sum$ imagens = 0,3 min X = 5,1 minutos
M5.3	Número de entradas explícitas	15
M5.4	Tempo Médio Entre Falhas	$T = 87$ minutos A = 28 X = 3

Os resultados coletados dessas medidas foram ruins. A medida M5.1 indica que a cada 6,2 minutos de utilização, o usuário precisa mudar de foco. As mudanças de foco foram: reiniciar a aplicação (12) e trocar de conexão com a rede (2).

A medida M5.2 indica que cerca de 5 minutos é considerado tempo perdido. Os tempos de adaptação e *logins* foram os piores tempos contabilizados. Já a medida M5.3 indica que muitas entradas explícitas são necessárias para interagir com a aplicação.

A medida M5.4 indica que a cada 3 minutos de uso uma falha na aplicação acontece, requisitando bastante atenção do usuário. Por fim, a medida M4.5 (Grau de Intervenção do

Usuário) apresentou que muitas vezes (50%) o usuário precisa intervir na adaptação para que ela ocorra.

Com esses resultados: alta frequência de mudanças de foco, alto tempo perdido, número muito grande entradas explícitas e intervenções e, por fim, muitas falhas durante a operação da aplicação, pode-se concluir que a aplicação GREat Tour requisita muito a atenção do usuário durante o uso.

### Questão 6 - Qual o grau de calma da interação?

Para essa sexta e última questão, as medidas M6.1, M6.2, M6.3 e M6.4 foram coletadas. Os resultados são descritos Tabela 5.21 e discutidos a seguir.

**Tabela 5.20** – Resultados GREat Tour – Questão 6

<b>ID da Medida</b>	<b>Nome da Medida</b>	<b>Resultado</b>
M6.1	Grau de disponibilidade	Alto = 0 <b>Médio = 3</b> Baixo = 2 Muito Baixo = 1
M6.2	Grau de relevância da interação	<b>Alto = 3</b> <b>Médio = 3</b> Baixo = 0 Muito Baixo = 0
M6.3	Grau de sincronia da interação	Alto = 2 <b>Médio = 4</b> Baixo = 0 Muito Baixo = 0
M6.4	Grau de cortesia da interação	<b>Alto = 3</b> Médio = 2 Baixo = 0 Muito Baixo = 1

Os resultados em geral foram bons. Uma medida ficou com dois resultados: M6.2, indicando que o grau de relevância da interação é alto para uns e médio para outros. É interessante notar que as três pessoas que responderam “Médio” são do grupo da computação e as outras pessoas que responderam “Alto” não são da computação.

## 5.5 Discussão dos Resultados

Com base nos resultados obtidos através da coleta das medidas, uma análise geral dos dados foi realizada de acordo com cada característica de qualidade (Sensibilidade ao Contexto, Mobilidade, Transparência, Atenção e *Calmness*)

Com relação à *Sensibilidade ao Contexto*, a aplicação UbiPrinter e UbiMute obteve um resultado muito abaixo do apropriado, pois os resultados das questões permitem visualizar que nem sempre as adaptações ocorrem quando há uma mudança contextual e quando essas adaptações ocorrem, muitas vezes, são incorretas de acordo com o real contexto do usuário, que por sua vez não é coletado de forma que seja mais confiável.

Para contornar esse problema, a aplicação UbiMute poderia levar em consideração duas informações de contexto adicionais: a localização atual do usuário e a localização do evento que ele cadastrou na agenda. Comparando essas duas informações é possível saber se o usuário está no local do evento que ele cadastrou e assim alterar o perfil do celular para o silencioso de forma mais confiável. Caso o usuário não estivesse em um raio aproximado do local do evento cadastrado, a aplicação não colocaria o celular no silencioso.

Já para a aplicação UbiPrinter, técnicas de localização indoor também se fazem necessárias (fusão de sensores, por exemplo.) para inferir com maior acurácia a localização do usuário dentro do prédio do GREat.

Somente a aplicação GREat Tour possui um valor aceitável para essa característica, pois apesar de utilizar informações de contexto de baixo nível, o grau de corretude das adaptações e do contexto foi muito alto.

Com relação à *Mobilidade*, a UbiPrinter, UbiMute e GREat Tour obtiveram resultados muito abaixo do apropriado, pois nenhum tipo de mobilidade é realizada nessas aplicações. Além disso, falhas de conexão ocorrem frequentemente e as aplicações não possuem alternativas para funcionar *off-line* enquanto o usuário não possui acesso a internet.

Com relação à *Transparência*, a UbiPrinter e a UbiMute obtiveram resultados muito satisfatórios. A aplicação UbiPrinter possui entradas implícitas para determinar a localização do usuário de modo que ele não sabe como a aplicação está realizando isso. A UbiMute é um serviço em segundo plano que não exige muita interação do usuário e é capaz de substituir duas ações comuns no cotidiano do usuário: colocar e retirar o dispositivo móvel no modo silencioso. Por outro lado, o GREat Tour obteve resultados muito abaixo do apropriado pois exige intervenções direta dos usuários para que a adaptação se realize.



Com relação à *Atenção*, a UbiPrinter apresenta resultados muito abaixo do satisfatório uma vez que a quantidade de mudança de foco foi alta (15), o que gerou perda de tempo para a tarefa principal (4,65 min) e causou várias falhas (TMEF = 1,8).

Para a UbiMute, o resultado obtido foi muito satisfatório, pois não foi necessário o usuário mudar de foco frequentemente, o tempo perdido foi somente o tempo de adaptação (63 ms), o número de entradas explícitas é baixo (2) e não houve tempo entre falhas.

Por outro lado, o GREat Tour obteve um resultado insatisfatório, pois muitas ações que tiraram o foco do usuário foram executadas (14), o tempo perdido é considerado alto (5,1 minutos), a aplicação possui muitas entradas explícitas e o tempo médio entre as falhas foi de 3 minutos. Assim, muita atenção é requisitada do usuário durante a interação.

Com relação à *Calmness*, característica que é medida a partir dos usuários, a UbiPrinter obteve um resultado satisfatório, provendo informações relevantes, quase sempre no momento correto. A aplicação UbiMute obteve um grau aceitável de *calmness* pelos usuários, provendo informações relevantes, no momento correto e substituindo ações. No entanto, os usuários acharam que o contexto não teve efeito sobre quando a interação foi oferecida (M6.3). Já na aplicação GREat Tour, o grau de *calmness* precisa ser melhorado, pois a maioria das medidas obteve resultado médio.

Assim, pode-se perceber que as aplicações ainda precisam se aperfeiçoar muito para serem consideradas aplicações ubíquas com qualidade de IHC. Se por um lado, a *Sensibilidade ao Contexto* obteve um bom resultado, por outro, a *Transparência* e, conseqüentemente, a *Atenção* são afetadas. Encontrar um equilíbrio entre essas características é o ideal para que a aplicação seja ubíqua e com qualidade de IHC.

Essa decisão pode ser baseada nos requisitos de cada aplicação. Uma aplicação de guia de visita, como o GREat Tour, exige bastante interação com usuário, diminuindo assim, sua transparência e exigindo atenção do usuário. No entanto, essa requisição de atenção pode ser minimizada através de técnicas para tornar a interação mais natural, por exemplo, a localização *indoor* pode ser inferida através da fusão de sensores (magnetômetro, acelerômetro e *wifi*) (UBEJD; ANGEL, 2011).

## 5.6 Considerações Finais

Este capítulo apresentou os resultados encontrados através da aplicação das medidas de *software* em três aplicações desenvolvidas no GREat (UbiPrinter, UbiMute e GREat Tour). No total doze pessoas participaram da avaliação da UbiPrinter. Dessa doze, oito utilizaram a UbiMute e seis utilizaram o GREat Tour.

Em geral, os resultados foram baixos para as características de *Sensibilidade ao Contexto*, *Mobilidade* e *Atenção*. Para *Transparência*, os resultados, em sua maioria, foram altos. Por último, a característica *Calma* obteve resultado satisfatório. Sendo assim, os resultados indicam que as aplicações precisam melhorar para se tornarem aplicações ubíquas com qualidade de IHC.

Portanto, a coleta dessas medidas de software possibilita avaliar a sua viabilidade, onde os resultados são um indício de que essas medidas são capazes de avaliar uma aplicação ubíqua conforme as características de qualidade de IHC selecionadas. No entanto, é importante ressaltar que outros estudos de casos com diferentes sistemas ubíquos são necessários para melhor avaliar a aplicabilidade dessas medidas e inferir se outras medidas são necessárias para avaliar as características de qualidade de IHC.

# 6 Conclusão

Este trabalho de dissertação propôs um conjunto de características e medidas de *software* para avaliação da qualidade da IHC em sistemas ubíquos. Este capítulo resume os resultados alcançados na Seção 6.1, apresenta os artigos produzidos durante o mestrado na Seção 6.2 e discute os trabalhos futuros que podem ser derivados do trabalho de dissertação na Seção 6.3.

## 6.1 Resultados Alcançados

Os sistemas ubíquos são capazes de monitorar os usuários, suas atividades e seu ambiente com o propósito de fornecer serviços relevantes, sem que o usuário perceba que está interagindo com uma infraestrutura computacional. Assim, a interação torna-se mais invisível ou com um mínimo de distração por parte do usuário.

As diferenças para os sistemas tradicionais (i.e., sensibilidade ao contexto, mudança na iniciativa da interação, diversificação de interfaces e mudanças nos propósitos) indicam a necessidade de adaptação nas avaliações da qualidade desses sistemas, incluindo a avaliação da qualidade da interação humano-computador.

Uma forma de avaliar a qualidade é realizando medições de *software*. Para executar essa atividade, é necessário conhecer muito bem o que se pretende medir. Para tanto, um Mapeamento Sistemático foi realizado para conhecer o que já estava sendo definido e avaliado na literatura e, assim, encontrar lacunas para a realização desse trabalho.

O objetivo do MS foi encontrar iniciativas relacionadas a características de qualidade, medidas de *software* e modelos de qualidade para IHC em sistemas ubíquos. Os resultados encontrados indicam que há uma grande lacuna na literatura dessa área, principalmente com relação a trabalhos que tenham sido aplicados na prática, isto é, com experimentos controlados ou estudos de casos.

Os resultados também expõem que as medidas encontradas não possuem uma definição completa. A grande maioria delas não apresenta propósito, funções de medições, valores de interpretação e métodos de coleta. Inviabilizando uma avaliação da qualidade com essas medidas.

Outro problema importante encontrado é que muitos trabalhos avaliam características pertinentes para qualquer tipo de sistema (*e.g.*, privacidade, segurança, usabilidade). Por outro

lado, como esperado, foram encontradas características específicas e essenciais para sistemas ubíquos (*e.g.*, sensibilidade ao contexto, mobilidade, *calmness*).

Entretanto, muitos trabalhos utilizam diferentes nomes para um mesmo conceito, indicando uma não padronização nas definições. Com o intuito de organizar e sumarizar todas as informações extraídas por esse mapeamento, uma análise semântica e uma revisão por pares foram realizadas para definir uma lista final de características de qualidade da IHC em sistemas ubíquos. No total, vinte e seis características foram definidas e, dessas, cinco foram escolhidas (Sensibilidade ao Contexto, Transparência, Mobilidade, Atenção e *Calmness*), para ser alvo das definições de medidas de software.

Essas cinco características foram escolhidas por serem específicas e essenciais somente para os sistemas ubíquos, entretanto, é importante salientar que a avaliação da qualidade da IHC envolve todas as características, como facilidade de uso, facilidade de aprendizado, entre outras.

Com as características de qualidade escolhidas, o método GQM foi utilizado para definição de medidas de *software*. Cinco reuniões foram realizadas para entrevistar especialistas em sistemas ubíquos. Tais especialistas são professores, pesquisadores e estudantes da pós-graduação no grupo de pesquisa GREat.

No total foram definidas vinte e quatro medidas de software que avaliam seis questões de um objetivo de medição. Essas medidas foram aplicadas em três estudos de caso com as aplicações GREat Tour, UbiPrinter e UbiMute. Como resultado desses estudos de caso, as medidas definidas indicam que essas aplicações ainda precisam evoluir bastante para serem consideradas ubíquas e com qualidade em IHC. Isso revela um indício de que as medidas propostas são capazes de avaliar as características de qualidade. No entanto, outros estudos de casos são necessários para avaliar a aplicabilidade dessas medidas.

Dado que essa é uma área de pesquisa com muitas lacunas a serem preenchidas, espera-se que esse trabalho seja uma de muitas contribuições do grupo de pesquisa em direção a uma abordagem de medição para avaliar a IHC em sistemas ubíquos.

## 6.2 Produção Bibliográfica

Nesta seção são apresentados os artigos publicados pela autora durante o período de mestrado, bem como a contribuição de cada artigo para o processo de aprendizado e realização dessa pesquisa. A lista é descrita a seguir.

- “Testes de Aplicações Móveis: uma análise das pesquisas científicas via revisão sistemática” Santos, I. S., Dantas, V. L. L., **Santos, R. M.** e Andrade, R. M. C. (aceito no SBQS 2012 – Qualis B3)
  - Contribuição: Participação em todo processo da revisão sistemática, o que contribuiu para realização do mapeamento sistemático desenvolvido desse trabalho.
- “Uma Avaliação de Ferramentas para Testes em Sistemas de Informação Móveis baseada no Método DMADV” Santos, I. S., Bezerra, C. I. M., Monteiro, G. S., Araújo, L. I., Oliveira, T. A., **Santos, R. M.**, Dantas, V. L. L. e Andrade, R. M. C. (aceito no SBSI 2013 – Qualis B4)
  - Contribuição: Escrita e Participação em uma das atividades do método DMADV.
- “Uma Proposta de Modelo de Qualidade para Avaliação da Interação Humano-Computador em Sistemas Ubíquos” **Santos, R. M.**, Andrade, R. M. C., Oliveira, K. M. (publicado no WTDQS/SBQS 2013 – Qualis B3) e
- “A Quality Model to Human-Computer Interaction Evaluation in Ubiquitous Systems” **Santos, R. M.**, Oliveira, K. M., Andrade, R. M. C., Santos, I. S. e Lima, E. R. (aceito no CLIHC 2013 – Qualis B5)
  - Contribuição: Esses últimos dois artigos estão diretamente relacionados com o tema de pesquisa.

## 6.3 Trabalhos Futuros

As perspectivas de trabalhos futuros e aprimoramento desse trabalho são descritas a seguir.

- Propor outras medidas de *software*

Outras medidas de *software* precisam ser definidas para avaliar as características de qualidade restantes encontradas no mapeamento, por exemplo, medidas para privacidade, segurança, usabilidade e até medidas para avaliar mais profundamente a qualidade do contexto utilizado pelas aplicações.

- Analisar dependência entre características

Verificar quanto o resultado da medição de uma característica pode impactar no resultado de outra característica, indicando assim uma dependência entre elas. Por exemplo, a sensibilidade ao contexto impacta na usabilidade da aplicação?

- Construir um modelo de qualidade

Organizar as características encontradas em subcaracterísticas e características, definir medidas para todas e construir um modelo de qualidade para sistemas ubíquos.

- Realizar medições com outros sistemas ubíquos

Para obter um valor de interpretação mais preciso, uma base histórica de dados de avaliações é necessária. Assim, torna-se essencial realizar medições várias vezes, com diferentes sistemas ubíquos para construir uma base de dados de avaliação, possibilitando, um aprimoramento na interpretação de um resultado.

- Facilitar a coleta de dados

Uma das dificuldades encontradas durante o estudo de caso é coletar os dados nos logs gerados. Neste trabalho, os logs não tinham uma padronização e as aplicações foram instrumentadas de forma que sua manutenção foi prejudicada. Criar uma ferramenta para facilitar a coleta de dados das interações envolvendo as informações de contexto que foram utilizadas e as adaptações utilizadas, bem como falhas ocorridas durante a interação seria essencial para facilitar a atividade de coleta e não aumentar a complexidade da aplicação a ser avaliada.

- Definir medidas para Interação Ator-Computador.

Nos sistemas ubíquos, a interação vai além da interação humano-computador, envolve também a interação ator-computador, onde ator pode ser um sistema, um dispositivo ou o próprio usuário. No caso da interação entre diferentes sistemas, a interoperabilidade é uma importante característica a ser avaliada através de medidas de *software*.

- Associar uma metodologia para avaliação da IHC em sistemas ubíquos

Avaliar somente através de questionários ou logs não é suficiente para obter todas as respostas necessárias para melhoria da aplicação. Assim, utilizar essas medidas subjetivas e outros métodos para coleta de dados (*e.g.*, testes, questionários, logs de interação e entrevistas) para criar uma metodologia capaz de avaliar a IHC nesses sistemas torna-se necessária..

# Referências Bibliográficas

ABI-CHAR, P. E. et al. **A Flexible Privacy and Trust Based Context-aware Secure Framework** Proceedings of the Aging Friendly Technology for Health and Independence, and 8th International Conference on Smart Homes and Health Telematics. **Anais...: ICOST'10**. Berlin, Heidelberg: Springer-Verlag, 2010. Disponível em: <<http://dl.acm.org/citation.cfm?id=1894439.1894443>>

ABOWD, G. D.; MYNATT, E. D.; RODDEN, T. The Human Experience [of Ubiquitous Computing]. **Pervasive Computing, IEEE**, v. 1, n. 1, p. 48–57, 2002.

ARAÚJO, R. B. DE. **Computação Ubíqua: Princípios, Tecnologias e Desafios** XXI Simpósio Brasileiro de Redes de Computadores. **Anais...2003**

BACHA, F.; OLIVEIRA, K.; ABED, M. Using Context Modeling and Domain Ontology in the Design of Personalized User Interface. **Internacional Journal on Computer Science and Information Systems (IJCSIS)**, 2011.

BARBOSA, S. D. J.; SILVA, B. S. **Interação Humano-Computador**. [s.l.] Elsevier, 2010.

BASILI, V.; ROMBACH, H. **Goal Question Metric Paradigm**. [s.l.] Encyclopedia of Software Engineering – 2, 1994. v. 1p. 528–532

BETTINI, C. et al. A Survey of Context Modelling and Reasoning Techniques. **Pervasive and Mobile Computing**, v. 6, n. 2, p. 161–180, 2010.

BROY, M. et al. **Formalizing the Notion of Adaptive System Behavior** Proceedings of the 2009 ACM Symposium on Applied Computing. **Anais...: SAC '09**. New York, NY, USA: ACM, 2009. Disponível em: <<http://doi.acm.org/10.1145/1529282.1529508>>

CAPPIELLO, I.; PUGLIA, S.; VITALETTI, A. Design and Initial Evaluation of a Ubiquitous Touch-Based Remote Grocery Shopping Process. **First International Workshop on Near Field Communication**, p. 9–14, fev. 2009.

CHAIRMAN-HEWETT, T. T. **ACM SIGCHI curricula for human-computer interaction**. [s.l.] ACM, 1992.

CHANG, Y.-H.; LIN, B.-S. **An Inquiry-based Ubiquitous Tour System** International Conference on Complex, Intelligent and Software Intensive Systems. **Anais...2011**

CHENG, B. H. C. et al. Software Engineering for Self-Adaptive Systems: A Research Roadmap. **LNCS**, p. 1–26, 2009.

DAMIÁN-REYES, P.; FAVELA, J.; CONTRERAS-CASTILLO, J. Uncertainty Management in Context-Aware Applications: Increasing Usability and User Trust. **Wireless Personal Communications**, v. 56, n. 1, p. 37–53, 2011.

DE MOOR, K. et al. Proposed Framework for Evaluating Quality of Experience in a Mobile, Testbed-oriented Living Lab Setting. **Mobile Networks and Applications**, v. 15, n. 3, p. 378–391, 2010.

DEY, A. K. Understanding and Using Context. **Personal and ubiquitous computing**, v. 5, n. 1, p. 4–7, 2001.

DEY, A. K.; ABOWD, G. D. **Towards a Better Understanding of Context and Context-Awareness** Relatório Técnico GIT-GVU-99-22, College of Computing, Georgia Institute of Technology. [s.l: s.n.].

DONOHOO, B. K.; OHLSEN, C.; PASRICHA, S. **AURA: An Application and User Interaction Aware Middleware Framework for Energy Optimization in Mobile Devices** IEEE International Conference on Computer Design: VLSI in Computers and Processors. **Anais...**2011

EVERS, V. et al. Interacting with adaptive systems. In: **Interactive Collaborative Information Systems**. [s.l.] Springer, 2010. p. 299–325.

FERNANDES, P. C. C. **UBIFEX: Uma Abordagem para Modelagem de Características de Linha de Produtos de Software Sensíveis ao Contexto**. [s.l.] Universidade Federal do Rio de Janeiro, 2009.

GARLAN, D.; SIEWIOREK, D. P.; STEENKISTE, P. Toward Distraction-Free Pervasive Computing. **Pervasive Computing, IEEE**, 2002.

GRIMM, R. et al. System Support for Pervasive Applications. **ACM Trans. Comput. Syst.**, v. 22, n. 4, p. 421–486, 2004.

GUERRA, A. C.; COLOMBO, R. M. T. Qualidade de Produto de Software. **PBQP/MCT. Disponível em:** < <http://www.mct.gov.br/index.php/content/view/2867.html#lista>>, 2009.

HAAPALAINEN, E. et al. **Psycho-physiological Measures for Assessing Cognitive Load** Proceedings of the 12th ACM International Conference on Ubiquitous Computing. **Anais...**: Ubicomp '10. New York, NY, USA: ACM, 2010 Disponível em: <<http://doi.acm.org/10.1145/1864349.1864395>>

HUGHES, B. **Practical Software Measurement**. [s.l.] McGraw - Hill Maidenhead, 2000.

IQBAL, R. et al. **User-Centred Design and Evaluation of Ubiquitous Services** Proceedings of the 23rd International Conference on Design of Communication - Documenting and Designing for Pervasive Information. **Anais...**2005 Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-32044471829&partnerID=40&md5=9a711a41b59e2b258fd13669a3826bc7>>

ISO 9241-11. Ergonomic Requirements for Office Work with Visual Display Terminals. **The International Organization for Standardization**, 1998.

ISO/IEC 14598. **Information Technology — Software Product Evaluation - Part 1** 1999



ISO/IEC 25000. Software Engineering - Software Product Quality Requirements and Evaluation (SQuaRE) – Guide to SQuaRE. 2005.

ISO/IEC 9126. Software Engineering – Product Quality – Part 1. 2001.

JAFARI, S. et al. **Privacy Metrics in Ubiquitous Computing Applications** International Conference for Internet Technology and Secured Transactions. **Anais...**2010

JIA, L.; COLLINS, M.; NIXON, P. **Evaluating Trust-Based Access Control for Social Interaction** 3rd International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies. **Anais...**2009 Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-77951440915&partnerID=40&md5=241ed20a5c57e71b94e32fcaec543a9e>>

KEMP, E. A.; THOMPSON, A.-J.; JOHNSON, R. S. **Interface Evaluation for Invisibility and Ubiquity: An Example from E-learning** Proceedings of the 9th ACM SIGCHI New Zealand Chapter's International Conference on Human-Computer Interaction: Design Centered HCI. **Anais...**: CHINZ '08. New York, NY, USA: ACM, 2008 Disponível em: <<http://doi.acm.org/10.1145/1496976.1496981>>

KIM, H. J.; CHOI, J. K.; JI, Y. **Usability Evaluation Framework for Ubiquitous Computing Device** Proceedings - 3rd International Conference on Convergence and Hybrid Information Technology, ICCIT 2008. **Anais...**2008 Disponível em: <<http://www.scopus.com/inward/record.url?eid=2-s2.0-57849159822&partnerID=40&md5=2965dfe3adcf63a35f173ca1fd4cb7e9>>

KITCHENHAM, B. A.; CHARTERS, S. **Guidelines for Performing Systematic Literature Reviews in Software Engineering** Technical Report. EBSE-2007-01, Keele University, **Anais...**2007

KO, I.-Y.; KOO, H.-M.; JIMENEZ-MOLINA, A. User-centric Web Services for Ubiquitous Computing. In: **Advanced Techniques in Web Intelligence-I**. [s.l.: s.n.].

KOSCIANSKI, A.; SOARES, M. DOS S. Qualidade de software. **São Paulo: Novatec**, 2006.

KOUROUTHANASSIS, P. E.; GIAGLIS, G. M.; KARAIKOS, D. C. **Delineating The Degree of “Pervasiveness” in Pervasive Information Systems: An Assessment Framework and Design Implications** Pan-Hellenic Conference on Informatics, PCI . **Anais...**ago. 2008

KRYVINSKA, N.; STRAUSS, C.; ZINTERHOF, P. **“Variated Availability” Approach to the Services Manageable Delivering** Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS). **Anais...**2011

LEE, J. et al. **A User-Centered Approach for Ubiquitous Service Evaluation: An Evaluation Metrics Focused on Human-System Interaction Capability** Asia-Pacific Conference, APCHI. **Anais...**2008

LEE, J.; YUN, M. H. **Usability Assessment for Ubiquitous Services: Quantification of the Interactivity in Inter-Personal Services** IEEE International Conference on Management of Innovation & Technology. **Anais...**jun. 2012

- LIAMPOTIS, N. et al. A Privacy Framework for Personal Self-Improving Smart Spaces. **International Conference on Computational Science and Engineering**, 2009.
- LIMA, E. R. R. et al. **GREat Tour: Um Guia de Visitas Móvel e Sensível ao Contexto** Webmedia 2013 - Workshop on Tools and Applications 2013. **Anais...**2013
- LIMA, F. F. DE P. **SysSu - Um Sistema de Suporte para Computação Ubíqua**. [s.l.] Dissertação de Mestrado - Universidade Federal do Ceará, 2011.
- LIMA, F. F. P. et al. **Uma Arquitetura Desacoplada e Interoperável para Coordenação em Sistemas Ubíquos** Software Components, Architectures and Reuse (SBCARS). **Anais...**2011
- MACHADO, M. P.; SOUZA, S. F. Métricas e Qualidade de software. **Departamento de Informática-Universidade Federal do Espírito Santo**, 2004.
- MARINHO, F. G. et al. **Uma Proposta de Arquitetura para Linhas de Produto de Software Aninhadas no Domínio de Aplicações Móveis e Sensíveis ao Contexto** IV Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software. **Anais...**2010
- MARINHO, F. G. et al. MobiLine: A Nested Software Product Line for The Domain of Mobile and Context-Aware Applications. **Science of Computer Programming**, maio 2012.
- NETO, P. A. M. S. et al. A Systematic Mapping Study of Software Product Lines Testing. **Information and Software Technology**, maio 2011.
- NIELSEN, J. **Usability Engineering**. [s.l.] Access Online via Elsevier, 1994.
- PARK, R. E.; GOETHERT, W. B.; FLORAC, W. A. **Goal-Driven Software Measurement. A Guidebook**. [s.l.] Software Engineering Institute, Carnegie Mellon University, 1996.
- PETERSEN, K. et al. **Systematic Mapping Studies in Software Engineering** Proceedings of the 12th International Conference on Evaluation and Assessment in Software Engineering. **Anais...: EASE'08**. Swinton, UK, UK: British Computer Society, 2008 Disponível em: <<http://dl.acm.org/citation.cfm?id=2227115.2227123>>
- POPPE, R.; RIENKS, R.; DIJK, B. VAN. Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications. **AI for Human Computing**, p. 234–250, 2007.
- POSLAD, S. **Ubiquitous Computing: Smart Devices, Environments and Interactions**. [s.l.] Wiley, 2009.
- PRESSMAN, R. S. **Software engineering: a practitioner's approach**. 7th. ed. [s.l.] McGraw-Hill, 2009.
- RAMOS, C. S. **Avaliação de Sistemas Legados**. [s.l.] Dissertação de Mestrado - Universidade Católica de Brasília, 2004.
- RAMOS, C. S.; OLIVEIRA, K. M.; ANQUETIL, N. **Legacy Software Evaluation Model for Outsourced Maintainer** Eighth European Conference on Software Maintenance and Reengineering. **Anais...**2004a

- RAMOS, C. S.; OLIVEIRA, K. M.; ANQUETIL, N. **Conhecendo Sistemas Legados através de Métricas de Software** III Simpósio Brasileiro de Qualidade de Software. **Anais...**2004b
- RANGANATHAN, A. et al. **Towards a Pervasive Computing Benchmark** International Conference on Pervasive Computing and Communications Workshops. **Anais...**2005
- RIEKKI, J.; ISOMURSU, P.; ISOMURSU, M. Evaluating the Calmness of Ubiquitous Applications. In: **Product Focused Software Process Improvement**. [s.l.: s.n.].
- ROCHA, L. S. **AdaptiveRME e AspectCompose: Um Middleware Adaptativo e um Processo de Composição Orientado a Aspectos para o Desenvolvimento de Software Móvel e Ubíquo**. [s.l.] Dissertação de Mestrado - Universidade Federal do Ceará, 2007a.
- ROCHA, L. S. **AdaptiveRME e AspectCompose: Um Middleware Adaptativo e um Processo de Composição Orientado a Aspectos para o Desenvolvimento de Software Móvel e Ubíquo**. [s.l.] MS thesis, Federalal University of Cear{á}, Fortaleza, Brazil, 2007 (in Portuguese), 2007b.
- ROCHA, L. S. et al. **Ubiquitous Software Engineering: Achievements, Challenges and Beyond** Brazilian Symposium on Software Engineering (in portuguese). **Anais...**set. 2011
- ROMÁN, M. et al. A Middleware Infrastructure for Active Spaces. **Pervasive Computing, IEEE**, 2002.
- ROSS, T.; BURNETT, G. Evaluating the Human–Machine Interface to Vehicle Navigation Systems as An Example of Ubiquitous Computing. **International Journal of Human-Computer Studies**, out. 2001.
- RUBIO, J. M. L.; BOZO, J. P. **Approach to a Quality Process for the Ubiquitous Software Development** Electronics, Robotics and Automotive Mechanics Conference. **Anais...**2007
- RUNESON, P.; HÖST, M. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. **Empirical Software Engineering**, 19 dez. 2008.
- SATYANARAYANAN, M. Pervasive computing: vision and challenges. **IEEE Personal Communications**, v. 8, n. 4, p. 10–17, 2001.
- SCHALKWYK, J. et al. Your Word is my Command: Google Search by Voice: A Case Study. In: **Advances in Speech Recognition**. [s.l.: s.n.].
- SCHMIDT, A. Implicit Human Computer Interaction Through Context. **Personal and Ubiquitous Computing**, jun. 2000.
- SCHOLTZ, J.; CONSOLVO, S. Toward a Framework for Evaluating Ubiquitous Computing Applications. **IEEE Pervasive Computing**, 2004.
- SGORA, A.; VERGADOS, D. D. Handoff prioritization and decision schemes in wireless cellular networks: a survey. **Communications Surveys Tutorials, IEEE**, v. 11, n. 4, p. 57–77, 2009.

- SOMMERVILLE, I. **Engenharia de Software**. 9ª Edição ed. [s.l.] Pearson Education - Br, 2011.
- SOUSA, B.; PENTIKOUSIS, K.; CURADO, M. UEF: Ubiquity Evaluation Framework. **Wired/Wireless Internet Communications**, 2011.
- SPÍNOLA, R. O.; MASSOLLAR, J.; TRAVASSOS, G. **Checklist to Characterize Ubiquitous Software Projects** Brazilian Symposium on Software Engineering. **Anais...**2007
- SUN, T.; DENKO, M. K. **Performance Evaluation of Trust Management in Pervasive Computing** International Conference on Advanced Information Networking and Applications, AINA. **Anais...**2008
- THOMPSON, S. G.; AZVINE, B. No Pervasive Computing Without Intelligent Systems. **BT technology journal**, 2004.
- TOCH, E. **Super-Ego: A Framework for Privacy-Sensitive Bounded Context-Awareness** ACM International Workshop on Context-Awareness for Self-Managing Systems. **Anais...**2011
- UBEJD, S.; ANGEL, R. **Indoor Positioning using Sensor-fusion in Android Devices** Student thesis, Kristianstad University Sweden, School of Health and Society, , 2011.
- VAN SOLINGEN, R.; BERGHOUT, E. **The Goal/Question/Metric Method: a practical guide for quality improvement of software development**. [s.l.] McGraw-Hill, 1999.
- VIANA, W. et al. PhotoMap: from location and time to context-aware photo annotations. **Journal of Location Based Services**, v. 2, n. 3, p. 211–235, set. 2008.
- VIEIRA, V. et al. Modelos e Processos para o desenvolvimento de Sistemas Sensíveis ao Contexto. **Jornadas de Atualização em Informática**, p. 381–431, 2009.
- WAGNER, S.; TOFTEGAARD, T.; BERTELSEN, O. **Requirements for an Evaluation Infrastructure for Reliable Pervasive Healthcare Research** International Conference on Pervasive Computing Technologies for Healthcare. **Anais...IEEE**, 2012
- WAIBEL, A. et al. Computers in the Human Interaction Loop. In: **Handbook of Ambient Intelligence and Smart Environments**. [s.l.: s.n.].
- WARNOCK, D. **A Subjective Evaluation of Multimodal Notifications** Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2011 5th International Conference on. **Anais...**2011
- WEIHONG-GUO, A. et al. Using Immersive Video to Evaluate Future Traveller Information Systems. **IET Intelligent Transport Systems**, 2008.
- WEISER, M. **The Computer for the 21st Century** Scientific American. **Anais...**1991
- WEISER, M.; BROWN, J. S. The Coming Age of Calm Technology. **Beyond calculation, Copernicus**, 1997.

WIERINGA, R. et al. Requirements Engineering Paper Classification and Evaluation Criteria: A Proposal and A Discussion. **Requirements Engineering**, 24 nov. 2005.

WU, C. L.; FU, L. C. Design and Realization of a Framework for Human-System Interaction in Smart Homes. **IEEE Transactions on Systems, Man, and Cybernetics Part A:Systems and Humans**, jan. 2012.

YAU, S. S.; WANG, Y.; KARIM, F. **Development of Situation-Aware Application Software for Ubiquitous Computing Environments**Computer Software and Applications Conference. **Anais...**2002

YE, J.; DOBSON, S.; MCKEEVER, S. Situation identification techniques in pervasive computing: A review. **Pervasive and Mobile Computing**, v. 8, n. 1, p. 36–66, 2012.

YU, P. et al. Application mobility in pervasive computing: A survey. **Pervasive and Mobile Computing**, v. 9, n. 1, p. 2–17, 2013.

ZHANG, Y. et al. **Adaptive Service Delivery for Mobile Users in Ubiquitous Computing Environments**International conference on Ubiquitous Intelligence and Computing. **Anais...**2006

## Apêndice A – Medidas de Software

As medidas apresentadas na tabela abaixo foram extraídas do MS apresentado no Capítulo 3 e foram utilizadas como referência durante as entrevistas GQM. Elas estão classificadas de acordo com as características de qualidade que são destinadas a avaliar.

**Tabela 0.1 – Medidas de Software Encontradas pelo Mapeamento Sistemático**

ID	Medidas de Software	Característica	Referência
M1	Variedade de informações contextuais suportadas	Sensibilidade ao Contexto	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M2	Grau de adaptação a mudanças do contexto	Sensibilidade ao Contexto	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M3	Grau de operação de proatividade do sistema	Sensibilidade ao Contexto	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M4	Número e diversidade de tecnologias de sensores e dispositivos de acesso	Sensibilidade ao Contexto	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M5	Grau de integração e coordenação entre artefatos de TI e tipos de dispositivos	Sensibilidade ao Contexto	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M6	O grau de serviços não esperados recebidos pela provisão de funções de serviço não requisitada: se as necessidades implícitas do usuário do serviço ubíquo são providas ou não (questionário e teste com usuário)	Sensibilidade ao Contexto	(LEE et al., 2008)
M7	Grau reconhecido de tempo perdido durante o uso do serviço ubíquo (questionário e teste com usuário)	Sensibilidade ao Contexto	(LEE et al., 2008)
M8	Taxa de mudanças apropriadas do conteúdo do serviço para as preferências e hábitos do usuário automaticamente	Sensibilidade ao Contexto	(LEE et al., 2008)
M9	Grau/Tempo requerido para que o usuário modifique o procedimento funcional do serviço para sua comodidade	Sensibilidade ao Contexto	(LEE et al., 2008)
M10	Grau de distância que o usuário deve mover adicionalmente para receber o serviço no lugar e tempo apropriado	Sensibilidade ao Contexto	(LEE et al., 2008)
M11	Grau de alcance dos espaços físicos para reconhecer o status do serviço	Sensibilidade ao Contexto	(LEE et al., 2008)
M12	Taxa de ocorrência de erros durante o uso do serviço	Sensibilidade ao Contexto	(LEE et al., 2008)
M13	Confiança: probabilidade do contexto ser sentido ou deduzido corretamente	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)

**Tabela 0.1** – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação)

ID	Medidas de Software	Característica	Referência
M14	Acurácia: porcentagem de erro de contextos sentidos ou inferidos	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M15	<i>Freshness</i> : tempo médio entre leitura de contextos	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M16	Determinação: área em que a informação de localização pode ser limitada	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M17	Redução do número de ações de configuração que o usuário tem de fazer para configurar o ambiente	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M18	Redução do número de vezes que o usuário foi perturbado ou incomodado por uma ação proativa feita pelo sistema. Medido com base no feedback do usuário	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M19	Redução no número de ações de configurações	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M20	Melhoria da <i>seamlessness</i> das interações	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M21	Facilidade de processos de recuperação de informação, controle de versões e arquivamento medida pelo feedback do usuário	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M22	Redução do número de passos que o usuário tem de tomar para obter alguma informação ou o número de parâmetros que o usuário tem de entrar na sua consulta	Sensibilidade ao Contexto	(RANGANATHAN et al., 2005)
M23	Precisão: descreve exatamente como as informações de contexto fornecido espelha a realidade	Sensibilidade ao Contexto	(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)
M24	Probabilidade de correteude: probabilidade de que um pedaço de informação de contexto está correta	Sensibilidade ao Contexto	(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)
M25	Acurácia e precisão do sensor	Sensibilidade ao Contexto	(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)
M26	Desempenho do sensor	Sensibilidade ao Contexto	(DAMIÁN-REYES; FAVELA; CONTRERAS-CASTILLO, 2011)
M27	Acurácia: $(\text{falso positivos} + \text{verdadeiros positivos}) / (\text{falsos positivos} + \text{verdadeiros positivos} + \text{falso negativos} + \text{verdadeiros negativos})$	Sensibilidade ao Contexto	(TOCH, 2011)

**Tabela 0.1 – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 1)**

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M28	Automação: (número de decisões tomadas automaticamente pelo número de decisões enviadas para o usuário) + (número de decisões tomadas automaticamente)	Sensibilidade ao Contexto	(TOCH, 2011)
M29	Usabilidade das modalidades de interação e distrações percebidas pelos usuários	Transparência	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M30	Grau de embarcamento dos artefatos físicos no ambiente	Transparência	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M31	Grau de conformidade ou mudanças evocadas para a arquitetura física existente	Transparência	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M32	Número e tipos de modalidades de interação e grau de apoio para as interações através de interfaces naturais (tangível, baseado em voz)	Transparência	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M33	Comparações de eficácia em diferentes dispositivos de entrada e saída	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M34	Percepção da transparência no sistema	Transparência	(EVERS et al., 2010)
M35	Compreensão do usuário da explicação do sistema	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M36	Eficácia das interações providas para controle do usuário do sistema	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M37	Correspondência entre o modelo contextual do sistema e da situação real	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M38	Adequação da ação	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M39	Adequação entre a ação do sistema e a ação que o usuário teria solicitado	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M40	Tempo de introduzir explicitamente informação de personalização	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M41	Tempo para o sistema de aprender e adaptar-se a preferência do usuário	Transparência	(SCHOLTZ; CONSOLVO, 2004)
M42	Nível de interação = grau de interação de primeiro plano entre um usuário e um espaço	Atenção	(SCHOLTZ; CONSOLVO, 2004)



**Tabela 0.1 – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 2)**

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M43	Número de vezes que o usuário precisa mudar o foco devido a tecnologia	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M44	Número de ações/visores o usuários necessita alcançar para checar o progresso de uma interação	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M45	Número de eventos não observados por um usuário em tempos aceitáveis	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M46	Percentual de tempo que o usuário gasta trocando entre dois focos	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M47	Carga de trabalho imposta para o usuário focar	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M48	Distração: Tempo tomado da tarefa principal	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M49	Degradação de desempenho na tarefa primária	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M50	Nível de frustração do usuário	Atenção	(SCHOLTZ; CONSOLVO, 2004)
M51	Distração visual significativa (carga de trabalho)	Atenção	(ROSS; BURNETT, 2001)
M52	Número de alternâncias de atenção	Atenção	(HAAPALAINEN et al., 2010)
M53	Grau de cobertura da rede móvel ou wireless	Mobilidade	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M54	Grau da Qualidade de Serviço	Mobilidade	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M55	Capacidade de migração dos serviços e/ou aplicações	Mobilidade	(KOUROUTHANASSIS; GIAGLIS; KARAIKOS, 2008)
M56	Tempo médio entre falhas / (Tempo médio entre falhas + tempo médio para reparar)	Disponibilidade	(KRYVINSKA; STRAUSSS; ZINTERHOF, 2011)
M57	Tempo para completar o processo de compras	Eficiência	(CAPPIELLO; PUGLIA; VITALETTI, 2009)
M58	Número de erros na leitura de tags	Eficiência	(CAPPIELLO; PUGLIA; VITALETTI, 2009)
M59	Tempo para completar uma tarefa	Eficiência	(SCHOLTZ; CONSOLVO, 2004)
M60	Velocidade do Desempenho: Medidas de tempo desde a interação do usuário até o feedback para o usuário	Eficiência	(SCHOLTZ; CONSOLVO, 2004)
M61	Taxa: Novos usuários/unidade de tempo	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M62	Proporção de usuários que utilizam uma aplicação ao invés de uma alternativa	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M63	Estatísticas de uso da tecnologia	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)

**Tabela 0.1 – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 3)**

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M64	Mudanças na produtividade	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M65	Custo/benefício percebido	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M66	Continuidade de uso	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M67	Quantidade de sacrifício do usuário	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M68	Disposição do usuário para a compra de tecnologia	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M69	Tempo típico gasto configurando e mantendo a tecnologia	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M70	Número de usuários alvo	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M71	Fonte de suprimento de tecnologia	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M72	Categorias de usuário em pós-implantação	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M73	Número de tarefas que o usuário pode completar, as quais não foram originalmente vislumbradas	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M74	Habilidade do usuário de se adaptar a medida que melhorias e funcionalidades são adicionadas	Aceitabilidade	(SCHOLTZ; CONSOLVO, 2004)
M75	Intenção para usar o sistema (Por exemplo: “Eu gostaria de usar o sistema de novo para tarefas similares”)	Aceitabilidade	(EVERS et al., 2010)
M76	Mudanças na produtividade ou desempenho	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M77	Mudança na qualidade do <i>output</i>	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M78	Mudança de comportamento: Tipo, frequência e duração	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M79	Disposição para modificar comportamento ou tarefas para utilizar a aplicação	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M80	Taxa de conforto de sistemas portáteis	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M81	Requisitos definidos fora das normas sociais do usuário	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M82	Avaliações estéticas de componentes do sistema	Utilidade	(SCHOLTZ; CONSOLVO, 2004)
M83	Utilidade percebida	Utilidade	(EVERS et al., 2010)
M84	Expressividade da política de segurança: Suporte a regras obrigatórias e discricionárias, sensibilidade ao contexto, lidar com a incerteza, resolução de conflitos	Segurança	(RANGANATHAN et al., 2005)

**Tabela 0.1** – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 4)

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M85	Controle do usuário sobre informações pessoais: privacidade do conteúdo, privacidade de identidade e privacidade de localização	Segurança	(RANGANATHAN et al., 2005)
M86	Discrição de mecanismos de segurança: % do tempo usado para interagir com o subsistema de segurança (por exemplo, autenticação) para auxiliar a tarefa principal	Segurança	(RANGANATHAN et al., 2005)
M87	Voltas da cabeça: número total por tarefas	Usabilidade	(RANGANATHAN et al., 2005)
M88	Movimento físico: % do tempo utilizado para movimentos auxiliares na tarefa principal	Usabilidade	(RANGANATHAN et al., 2005)
M89	Prévio conhecimento do usuário: número total de fatos que devem ser conhecidos pelo usuário para executar uma tarefa	Usabilidade	(RANGANATHAN et al., 2005)
M90	Teclas digitadas, cliques e outras entradas atômicas: número total por tarefa	Usabilidade	(RANGANATHAN et al., 2005)
M91	Número total de erros e tempo gasto recuperando o erro	Usabilidade	(RANGANATHAN et al., 2005)
M92	Número mínimo de operações do usuário necessário para executar uma função particular	Usabilidade	(ROSS; BURNETT, 2001)
M93	Tempo médio para inserir um destino	Usabilidade	(ROSS; BURNETT, 2001)
M94	Tipo de informação que o usuário precisa divulgar para obter um resultado da aplicação	Privacidade	(SCHOLTZ; CONSOLVO, 2004)
M95	Disponibilidade de informação do usuário para outros usuários do sistema ou terceiros	Privacidade	(SCHOLTZ; CONSOLVO, 2004)
M96	Facilidade de coordenação com outros uma aplicação multi-usuário	Confiança	(SCHOLTZ; CONSOLVO, 2004)
M97	Número de colisões com a atividade de outros	Confiança	(SCHOLTZ; CONSOLVO, 2004)
M98	Entendimento do usuário sobre como os dados gravados são utilizados	Confiança	(SCHOLTZ; CONSOLVO, 2004)
M99	Compreensão do usuário sobre inferências que podem ser feitas sobre ele pela aplicação	Confiança	(SCHOLTZ; CONSOLVO, 2004)
M100	Habilidade de gerenciar como e por quem seus dados serão utilizados	Confiança	(SCHOLTZ; CONSOLVO, 2004)

**Tabela 0.1** – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 5)

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M101	Tipos de recurso disponíveis ao usuário no caso de seus dados serem utilizados abusivamente	Confiança	(SCHOLTZ; CONSOLVO, 2004)
M102	Eficácia de interações com um grande número de entidades ou usuários	Escalabilidade	(SCHOLTZ; CONSOLVO, 2004)
M103	Taxa de usuários que executam as tarefas	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M104	Nível de diversão ao utilizar uma aplicação	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M105	Nível de antecipação antes de utilizar a aplicação	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M106	Sensação de perda quando o aplicativo está indisponível	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M107	Estética: Classificação da aparência da aplicação	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M108	Orgulho em usar e possuir o aplicativo	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M109	Pressão sentida de outros para utilizar ou possuir a aplicação	Satisfação do Usuário	(SCHOLTZ; CONSOLVO, 2004)
M110	Escala de satisfação do usuário (1-5, no qual 5 é o valor de maior concordância)	Satisfação do Usuário	(RANGANATHAN et al., 2005)
M111	Largura de banda	Capacidade da Rede	(ZHANG et al., 2006)
M112	Latência	Capacidade da Rede	(ZHANG et al., 2006)
M113	Tipo de acesso atual (GPRS, UMTS, H SPA, LTE, Wi-Fi, WiMA X, DV B-H)	Capacidade da Rede	(DE MOOR et al., 2010)
M114	A força do sinal percebido	Capacidade da Rede	(DE MOOR et al., 2010)
M115	<i>Throughput</i>	Capacidade da Rede	(DE MOOR et al., 2010)
M116	<i>Delay</i>	Capacidade da Rede	(DE MOOR et al., 2010)
M117	<i>Jitter</i>	Capacidade da Rede	(DE MOOR et al., 2010)
M118	<i>Packetloss</i>	Capacidade da Rede	(DE MOOR et al., 2010)
M119	SO do dispositivo	Capacidade do Dispositivo	(DE MOOR et al., 2010)
M120	Utilização de CPU	Capacidade do Dispositivo	(DE MOOR et al., 2010)

**Tabela 0.1 – Medidas de Software Encontrados pelo Mapeamento Sistemático (Continuação 6)**

<b>ID</b>	<b>Medidas de Software</b>	<b>Característica</b>	<b>Referência</b>
M121	Consumo de memória	Capacidade do Dispositivo	(DE MOOR et al., 2010) (ZHANG et al., 2006)
M122	Tempo de Vida e status da bateria	Capacidade do Dispositivo	(DE MOOR et al., 2010) (ZHANG et al., 2006)
M123	Tamanho da tela	Capacidade do Dispositivo	(ZHANG et al., 2006) (DE MOOR et al., 2010)
M124	Intensidade da Cor	Capacidade do Dispositivo	(ZHANG et al., 2006)
M125	Concordância do serviço: A(número de funções executadas e esperadas)/B(número de funções providas pelo serviço ubíquo)	-	(LEE; YUN, 2012) (LEE et al., 2008)
M126	Grau de participação do usuário para uma comunicação bidirecional dos serviços ubíquos: questionários e teste com o usuário	-	(LEE; YUN, 2012) (LEE et al., 2008)
M127	Grau de imersão do usuário no serviço ubíquo: questionário e teste com o usuário	-	(LEE; YUN, 2012) (LEE et al., 2008)
M128	Grau de entendimento: A (número de entradas e saídas executadas e esperadas)/ B (número de entradas e saídas providas pelo serviço ubíquo)	-	(LEE; YUN, 2012) (LEE et al., 2008)
M129	Tempo de aprovação do usuário antes de usar o serviço	Experiência do Usuário	(LEE et al., 2008)
M130	Tempo de aprendizado para utilizar novas funções do serviço	Experiência do Usuário	(LEE et al., 2008)
M131	Tempo gasto em hesitação ou espera para utilizar o serviço	Experiência do Usuário	(LEE et al., 2008)
M132	Número de usuários fora do controle durante o uso do sistema	Experiência do Usuário	(LEE et al., 2008)
M133	Porcentagem de faltas transientes que foram invisíveis para o usuário	Robustez	(SCHOLTZ; CONSOLVO, 2004)
M134	Volatilidade: Medidas de interrupção baseadas no conjunto dinâmico de usuário, <i>hardware</i> ou <i>software</i>	Robustez	(SCHOLTZ; CONSOLVO, 2004)
M135	Previsibilidade do comportamento da aplicação: grau de compatibilidade entre modelo do usuário e o comportamento da aplicação.	-	(SCHOLTZ; CONSOLVO, 2004)
M136	Grau de compatibilidade entre o modelo do usuário e a funcionalidade da aplicação.	-	(SCHOLTZ; CONSOLVO, 2004)
M137	Grau de compatibilidade entre o entendimento do usuário sobre suas responsabilidades, responsabilidades do usuário e a situação real.	-	(SCHOLTZ; CONSOLVO, 2004)

**Tabela 0.1** – Medidas de *Software* Encontrados pelo Mapeamento Sistemático (Continuação 7)

<b>ID</b>	<b>Medidas de <i>Software</i></b>	<b>Característica</b>	<b>Referência</b>
M138	Grau que o usuário entende os limites da aplicação	-	(SCHOLTZ; CONSOLVO, 2004)
M139	Noção de vocabulário: grau de compatibilidade entre o modelo do usuário e a sintaxe usada pela aplicação.	-	(SCHOLTZ; CONSOLVO, 2004)
M140	Método de entrada utilizável ou vários graus de interatividade (teclado, caneta stylus, entrada de áudio, tela de toque, etc.)= Número médio ou grau (questionário e teste com o usuário) - Tipo, grau aplicável de interface multimodal	-	(LEE; YUN, 2012)
M141	Grau de usabilidade de uma mesma funcionalidade implementada em dispositivos multimodais diferentes: Grau aplicável da mesma função com interface multimodal (questionário e teste com o usuário)	-	(LEE; YUN, 2012)
M142	Faixa de requisição de serviço utilizável (distância sentida) = distância da cobertura do serviço - distância bem sucedida baseada na localização do usuário – grau reconhecível de uso para uma distância utilizável (questionário e teste com usuário)	-	(LEE; YUN, 2012)
M143	Proporção adequada do número de funções de serviço de ocorrências de problema ou o número total de funções.	-	(LEE; YUN, 2012)
M144	Grau de ajuste de dados entre serviços em operação e componentes de outros serviços.	-	(LEE; YUN, 2012)
M145	Grau de conhecimento entre o serviço de defesa do sistema contra a invasão de outro usuário	-	(LEE; YUN, 2012)
M146	Grau de entendimento instintivo da função de serviços resultantes.	-	(LEE; YUN, 2012)
M147	Frequência do número de erros de sistema durante o uso do dispositivo.	-	(LEE; YUN, 2012)
M148	Número de requisições de ajuda para confirmar o resultado de saída do dispositivo.	-	(LEE; YUN, 2012)
M149	Grau de <i>feedback</i> sentido pelo usuário.	-	(LEE; YUN, 2012)
M150	Tempo requerido até a finalização do carregamento de uma função específica.	-	(LEE; YUN, 2012)
M151	Taxa de Erros de Palavra: (número de substituições + inserção + deleções)/número total de palavras.	-	(SCHALKWYK et al., 2010)

**Tabela 0.1** – Medidas de *Software* Encontrados pelo Mapeamento Sistemático (Continuação 8)

M152	Qualidade Semântica: número de resultados de busca corretos sobre o número total de buscas faladas.	-	(SCHALKWYK et al., 2010)
M153	Taxa de Palavras Fora de Vocabulário: porcentagem de palavras faladas pelo usuário que não são modeladas pelo modelo de linguagens. Importante manter o número mais baixo possível.	-	(SCHALKWYK et al., 2010)
M154	Latência é definida como o tempo total, em segundos, levado para completar a busca requisitada por voz. Mais precisamente, nós definimos latência como o tempo no qual o usuário finaliza a fala até os resultados aparecerem na tela.	-	(SCHALKWYK et al., 2010)
M155	Perplexidade é, cruamente, uma medida de tamanho do conjunto de palavras que podem ser reconhecida em seguida, dada previamente a busca de palavras reconhecíveis.	-	(SCHALKWYK et al., 2010)

## Apêndice B – Classificação dos Estudos

A classificação dos estudos finais do MS nas facetas definidas para criar o mapa sistemático está apresentada na tabela a seguir.

**Tabela 0.2 - Classificação dos Estudos do Mapeamento Sistemático**

<b>Estudo</b>	<b>Faceta Contribuição</b>	<b>Faceta Tipo de Pesquisa</b>
(WU; FU, 2012)	<i>Framework</i> de Medidas	Proposta de Solução
(SOUSA; PENTIKOUSIS; CURADO, 2011)	<i>Framework</i> de Características	Proposta de Solução
(ABI-CHAR et al., 2010)	Questões de Qualidade	Proposta de Solução
(JIA; COLLINS; NIXON, 2009)	Questões de Qualidade	Proposta de Solução
(CAPPIELLO; PUGLIA;	<i>Framework</i> de Medidas	Proposta de Solução
(KIM; CHOI; JI, 2008)	<i>Framework</i> de Características	Proposta de Solução
(KOUROUTHANASSIS;	<i>Framework</i> de Medidas	Proposta de Solução
(SUN; DENKO, 2008)	Questões de Qualidade	Proposta de Solução
(LEE et al., 2008)	Modelo de Qualidade	Proposta de Solução
(WEIHONG-GUO et al., 2008)	Questões de Qualidade	Pesquisa de Validação
(RUBIO; BOZO, 2007)	Questões de Qualidade	Proposta de Solução
(ZHANG et al., 2006)	<i>Framework</i> de Medidas	Proposta de Solução
(RANGANATHAN et al., 2005)	<i>Framework</i> de Medidas	Proposta de Solução
(IQBAL et al., 2005)	Questões de Qualidade	Proposta de Solução
(SCHOLTZ; CONSOLVO, 2004)	Modelo de Qualidade	Artigos Filosóficos



**Tabela 0.2** – Classificação dos Estudos do Mapeamento Sistemático (Continuação)

(ROSS; BURNETT, 2001)	<i>Framework</i> de Medidas	Artigos de Opinião
(LEE; YUN, 2012)	<i>Framework</i> de Medidas	Proposta de Solução
(WAGNER; TOFTEGAARD;	<i>Framework</i> de Características	Proposta de Solução
(KEMP; THOMPSON;	<i>Framework</i> de Características	Proposta de Solução
(JAFARI et al., 2010)	<i>Framework</i> de Medidas	Artigos de Opinião
(CHANG; LIN, 2011)	Questões de Qualidade	Proposta de Solução
(TOCH, 2011)	<i>Framework</i> de Medidas	Pesquisa de Validação
(HAAPALAINEN et al., 2010)	<i>Framework</i> de Medidas	Pesquisa de Validação
(LIAMPOTIS et al., 2009)	Questões de Qualidade	Proposta de Solução
(KRYVINSKA; STRAUSS;	<i>Framework</i> de Medidas	Proposta de Solução
(DAMIÁN-REYES; FAVELA;	<i>Framework</i> de Medidas	Pesquisa de Validação
(EVERS et al., 2010)	<i>Framework</i> de Medidas	Pesquisa de Validação
(WAIBEL et al., 2010)	<i>Framework</i> de Medidas	Pesquisa de Validação
(DE MOOR et al., 2010)	<i>Framework</i> de Medidas	Proposta de Solução
(THOMPSON; AZVINE, 2004)	<i>Framework</i> de Características	Artigos de Opinião
(KO; KOO; JIMENEZ-	<i>Framework</i> de Características	Proposta de Solução
(SCHALKWYK et al., 2010)	<i>Framework</i> de Medidas	Pesquisa de Avaliação

# Apêndice C – Formulário para Definição do Perfil do Usuário

1. Nome

2. Sexo

(1) Feminino (2) Masculino

3. Idade

4. Ocupação Profissional

5. Formação Acadêmica

6. Já utilizou celulares com telas *touch screen*?

7. Qual seu grau de experiência no uso de celulares com telas *touch screen*?

(1)Iniciante (2)Experiente (3)Muito Experiente (4)Nenhuma

8. Já utilizou alguma aplicação para o celular que utiliza *QR Codes*?

(1)Sim (2)Não (3)Não sei

9. Já utilizou aplicações ubíquas (aplicações móveis e sensíveis ao contexto)?

(1)Sim (2)Não (3)Não sei (4)Não sei, não conheço o termo “ubíqua”

10. Qual sua experiência com aplicações ubíquas?

(1)Iniciante (2)Experiente (3)Muito Experiente (4)Nenhuma (5)Não sei

11. Já utilizou guias de visitas móveis e sensíveis ao contexto?

(1)Sim (2)Não (3)Não sei

12. Você já desenvolveu softwares?

(1)Sim (2)Não

# Apêndice D – Formulário para Coleta de Dados Após o Uso da Aplicação

## 1. A aplicação estava disponível quando você necessitou?

ALTO: A aplicação estava sempre disponível para interagir quando era necessário. Você conseguiu executar normalmente a função desejada sem nenhuma dificuldade.

MÉDIO: Geralmente estava disponível quando era necessário. Você conseguiu executar a aplicação, porém com dificuldades.

BAIXO: Disponível somente em alguns locais. Você conseguiu executar a aplicação com muita dificuldade.

MUITO BAIXO: A disponibilidade não corresponde com sua necessidade de onde e quando utilizar a aplicação. Você não foi capaz de executar funções desejadas.

## 2. A aplicação fornece somente informações relevantes?

ALTO: nenhuma informação ou requisição irrelevante é apresentada. Você obteve somente informações e requisições necessárias.

MÉDIO: a maioria das informações e requisições são relevantes.

BAIXO: informações e requisições irrelevantes foram frequentemente apresentadas.

MUITO BAIXO: muita informação é apresentada. Muitas opções de comando irrelevantes são apresentadas. Você precisou procurar comandos ou opções corretas em hierarquias de menu ou em uma longa lista de opções. Seu contexto (localização, atividades) não teve nenhum efeito sobre a informação ou escolha apresentada.

## 3. A aplicação interage no momento correto?

ALTO: a interação ocorreu sempre no momento correto. Quando a interação ocorreu, você concordou que foi no momento correto e o você não quis interagir quando ela não ocorreu.

MÉDIO: a interação ocorreu na maioria das vezes no momento correto. Poucas vezes aconteceu de você não querer a interação quando ela ocorreu, ou de querer interagir quando não ocorreu.

BAIXO: Na maioria das vezes, você não concordou com a interação.

MUITO BAIXO: O seu contexto (localização, atividades e preferências) não teve efeito sobre quando a interação foi oferecida.

## 4. A interação ocorre da melhor maneira possível?

ALTO: a aplicação apoiou ou substituiu suas ações. A aplicação requisitou sua atenção, somente quando necessário. A interação ocorreu de forma que você julga natural.

MÉDIO: a aplicação exige desnecessariamente a sua atenção, mas o incômodo não foi grave. A aplicação poderia não requisitar tanta atenção, mas você foi capaz de continuar suas atividades normalmente.

BAIXO: você precisou mudar o foco de atenção de suas atividades para a aplicação.

MUITO BAIXO: Você precisou interromper sua atividade em andamento para se concentrar completamente na interação com a aplicação.

**\*Perguntas Específicas para a UbiMute\***

5. Quando a aplicação colocou o celular no silencioso, você realmente estava no evento que tinha cadastrado na agenda? Se não, especifique quais eventos você cadastrou e não compareceu.
6. Você percebeu o seu celular ficando no silencioso por causa da aplicação UbiMute?
7. Você não atendeu alguma ligação enquanto o celular estava no silencioso?
8. A aplicação falhou em algum momento?

**\*Perguntas Específicas para a UbiPrinter\***

9. Você consegue identificar como a aplicação fez para descobrir sua localização? Se sim, descreva.
10. Você conseguiria alterar essas informações? Se sim, como?

# Apêndice E – Formulário de Avaliação para o Desenvolvedor

1. Qual tipo de informação de contexto é utilizada na aplicação?  
(1) Informação Crua (2) Situações de Contexto (3) Informações de Alto Nível
2. Descreva as informações de contexto cruas, as situações de contexto e as informações de alto nível utilizadas na aplicação
3. A aplicação utiliza alguma técnica para raciocinar sobre incertezas de informações de contexto?  
(1) Fuzzy Logic (2) Probabilistic Logic (3) Bayesian Networks (4) Hidden Markov Models (5) The Dempster-Shafer Theory of Evidence (6) Requisição para o Usuário (7) Nenhuma
4. A aplicação exige alguma intervenção explícita do usuário para a adaptação ocorrer?  
(1) Sim (2) Não
5. Quais intervenções o usuário pode ou precisa fazer?
6. Quais sensores são utilizados?
8. Para quais versões do Android a aplicação funciona?
9. Para quais dispositivos móveis a aplicação será executada?
10. Qual a quantidade de tecnologias de acesso a comunicação a aplicação tem acesso, descreva quais (wifi, 3G.)?
11. A aplicação possui entradas implícitas?  
(1) Sim (2) Não
12. Qual a quantidade de entradas implícitas da aplicação?
13. Descreva as entradas implícitas que a aplicação utiliza
14. Um usuário leigo da aplicação é capaz de perceber essas entradas implícitas?
15. Quais entradas implícitas o usuário pode perceber?
16. A entrada implícita pode ser manipulada pelo o usuário?
17. Quais entradas implícitas o usuário pode manipular?
18. A aplicação substitui ações do usuário?  
(1) Sim (2) Não
19. Quais ações são substituídas pela aplicação?
20. Quantas e quais ações o usuário precisa fazer para a aplicação inicializar?
21. A aplicação possui interfaces para a interação natural como baseada em voz, gestos ou outros?

## Apêndice F – Formulário de Observação para o Avaliador

<b>Data</b>		<b>Aplicação Utilizada</b>	
<b>Nome do Usuário</b>		<b>Cenário de Uso</b>	
<b>Horário de início da interação</b>		<b>Horário de término da interação</b>	
<b>Localização do usuário</b>			
<b>Resultado da Adaptação</b>			
<b>Mídias apresentadas</b>			
<b>Nível da bateria do dispositivo</b>			
<b>Falhas ocorridas durante a interação</b>			
<b>Distrações Ocorridas</b>			