



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

**Dissertação de Mestrado**

**Avaliação da Confiança no Funcionamento de Sistemas Ubíquos  
usando Medidas de Qualidade**

**Andressa Bezerra Ferreira**

Fortaleza – Ceará  
2015

**Andressa Bezerra Ferreira**

**Avaliação da Confiança no Funcionamento de Sistemas  
Ubíquos usando Medidas de Qualidade**

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.

Orientador: Rossana Maria de Castro  
Andrade, PhD.

Co-orientador: Reinaldo Bezerra Braga,  
Dr.

Fortaleza – Ceará  
2015

# Avaliação da Confiança no Funcionamento de Sistemas Ubíquos Usando Medidas de Qualidade

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

Aprovado em \_\_\_\_/\_\_\_\_/\_\_\_\_

## Banca Examinadora

---

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

---

Prof. Reinaldo Bezerra Braga, Dr. (Co-Orientador)  
Instituto Federal do Ceará – IFCE

---

Profa. Ana Regina Cavalcanti da Rocha, Dra.  
Universidade Federal do Rio de Janeiro – UFRJ

---

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

---

Prof. Antônio Mauro Barbosa de Oliveira, Dr.  
Instituto Federal do Ceará – IFCE

Ao Deus da minha estranha calma.

# Agradecimentos

*A Deus,*

A Ti Jesus, Meu Senhor e Meu Salvador!

Pela fidelidade e amor extremo.

*A minha família,*

Pelo amor, pelo incentivo, pela dedicação e por todos os sacrifícios que fizeram por mim.

*A minha orientadora Rossana,*

Por acreditar em mim. Por me dar a oportunidade de realizar este trabalho e insistir, revisar, cortar, corrigir, melhorar, acrescentar, sugerir e me incentivar a cuidar de cada parágrafo como se fosse o último, de cada capítulo como se fosse o único. Obrigada!

*Ao meu coorientador Reinaldo,*

Por buscar sempre enriquecer este trabalho com suas sugestões e críticas brilhantes e sinceras.

*Aos demais professores e profissionais envolvidos,*

Pelo generoso auxílio em compartilhar comigo os seus conhecimentos.

*Aos meus amigos,*

“Por diante da vastidão do tempo e imensidão do universo, me darem a honra de dividir um mesmo planeta, tempo e época com eles”.

*Ao MDCC, GREat, CTQS e ARiDa,*

Pela acolhida e por me darem quatro novas famílias.

Que sorte a minha ter vocês!

*A CAPES, FUNCAP e FCPC,*

Pelo apoio financeiro ao longo do mestrado.

“Talvez seja porque meus maiores desafios tenham vindo de ameaças externas, mas eu não acredito no banimento dos laços. Não posso imaginar a minha vida sem Mara, Leia, ou mesmo Han.”

Luke Skywalker

# Resumo

Sistemas ubíquos são, por princípio, autônomos e capazes de perceber o ambiente, as atividades e necessidades do usuário, sem que seja necessário determiná-las explicitamente. Conseqüentemente, a aceitação e utilização desse tipo de sistema requer confiança no funcionamento por parte do usuário. Uma das possíveis técnicas para a avaliação da confiança no funcionamento de um software é a realização de medições. Entretanto, apesar da confiança no funcionamento de um sistema ubíquo ser uma questão importante, por englobar diversos atributos de qualidade, garanti-la é uma tarefa difícil, devido as peculiaridades (e.g., sensibilidade ao contexto, mobilidade e heterogeneidade) inerentes aos sistemas ubíquos, o que torna a avaliação de qualidade mais desafiadora do que em sistemas convencionais. Sendo assim, este trabalho de dissertação de mestrado tem como objetivo investigar e, quando necessário, adaptar e propor medidas de qualidade de *software* para avaliação da confiança no funcionamento em sistemas ubíquos, com foco em aplicações executando em dispositivos móveis (e.g., *smartphones*, *tablets*, e *smart devices* em geral). Três estudos de casos são utilizados neste trabalho para avaliar as medidas definidas: um guia de visitas móveis, um sistema para controle de volume de chamada do celular, e um sistema de detecção e alerta de quedas. Como principal contribuição espera-se auxiliar o analista de qualidade na identificação de quais atributos estão impactando a confiança do usuário no funcionamento do sistema ubíquo, impulsionando, conseqüentemente, melhorias que possibilitem o aumento da aceitação e utilização desse tipo de sistema.

**Palavras-chave:** Sistemas Ubíquos, Qualidade de Software, Medidas de Qualidade, Confiança no Funcionamento, Confiança

# Abstract

Ubiquitous systems are, in principle, autonomous and capable of perceiving environment, activities and user needs, without having to explicitly determine them. Consequently, in order to use this type of systems, a user needs to trust them. One of the possible techniques to evaluate the user's trust in the system's functionalities is through measurements. However, despite the relevance of the user's trust in a ubiquitous system, since several quality attributes are involved, to assure that it is a hard task due to its specific characteristics such as context-awareness, mobility, heterogeneity what make its quality evaluation more challenging than in traditional systems. With that in mind, this master thesis' objective is to investigate and, when necessary, to adapt and propose software quality measures to evaluate the user's trust in the functionality of ubiquitous systems that run in mobile devices (e.g., *smartphones*, *tablets*, e *smart devices* in general). Three case studies are used in this work to evaluate the measures: a mobile visit guide, a cellphone's call volume control system, and an alert and detection system for falls. In short, the expected contribution of this work is to help software quality analysts in the identification of which attributes are impacting the trust of the user in the functionality of a ubiquitous system, and consequently, suggesting improvements to the developers that allow an increase in the acceptance and use of this type of system.

**Key-words:** Software quality, Ubiquitous Systems, Measures, User's Trust, Dependability.



# Sumário

<b>LISTA DE FIGURAS .....</b>	<b>12</b>
<b>LISTA DE TABELAS.....</b>	<b>14</b>
<b>LISTA DE ABREVIATURAS.....</b>	<b>15</b>
<b>CAPÍTULO 1.....</b>	<b>16</b>
<b>INTRODUÇÃO .....</b>	<b>16</b>
1.1    CONTEXTUALIZAÇÃO E MOTIVAÇÃO .....	16
1.2    OBJETIVOS E CONTRIBUIÇÕES ESPERADAS .....	19
1.3    METODOLOGIA UTILIZADA .....	20
1.4    ORGANIZAÇÃO DA DISSERTAÇÃO.....	21
<b>CAPÍTULO 2.....</b>	<b>23</b>
<b>REFERENCIAL TEÓRICO .....</b>	<b>23</b>
2.1    SISTEMAS UBÍQUOS .....	23
2.2    QUALIDADE DE SOFTWARE .....	28
2.3    CONFIANÇA NO FUNCIONAMENTO DE SISTEMAS DE SOFTWARE .....	36
2.4    CONSIDERAÇÕES FINAIS .....	42
<b>CAPÍTULO 3.....</b>	<b>45</b>
<b>TRABALHOS CORRELATOS .....</b>	<b>45</b>
3.1    METODOLOGIA DE BUSCA.....	45
3.2    ANÁLISE DOS TRABALHOS CORRELATOS .....	47
3.3    CONSIDERAÇÕES FINAIS .....	52
<b>CAPÍTULO 4.....</b>	<b>55</b>
<b>MEDIDAS DE QUALIDADE DE SOFTWARE .....</b>	<b>55</b>
4.1    PLANEJAMENTO .....	55
4.2    DEFINIÇÃO .....	57
4.3    COLETA DAS MEDIDAS.....	76
4.4    INTERPRETAÇÃO DOS RESULTADOS .....	76
4.5    CONSIDERAÇÕES FINAIS .....	77
<b>CAPÍTULO 5.....</b>	<b>81</b>
<b>ESTUDOS DE CASO.....</b>	<b>81</b>
5.2    GREAT MUTE .....	85
5.2.1  GREAT MUTE v 1.0.....	88
5.2.2  GREAT MUTE v 2.0 .....	91
5.2.3  GREAT MUTE v 3.0 .....	95
5.3    FALERT .....	98
5.3.1 <i>Mecanismo para a Detecção de Quedas</i> .....	101
5.3.2 <i>Testes Realizados</i> .....	102
5.3.3 <i>Interpretação dos Resultados</i> .....	107
5.4    GREAT TOUR.....	113
5.5    CONSIDERAÇÕES FINAIS .....	118
<b>CAPÍTULO 6.....</b>	<b>121</b>
<b>CONCLUSÃO .....</b>	<b>121</b>
5.6    RESULTADOS E CONTRIBUIÇÕES .....	121
5.7    TRABALHOS FUTUROS .....	124
<b>REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>126</b>
<b>APÊNDICE A.....</b>	<b>134</b>

<b>APÉNDICE B.....</b>	<b>180</b>
<b>APÉNDICE C.....</b>	<b>181</b>
<b>APÉNDICE D.....</b>	<b>184</b>

# Lista de Figuras

Figura 1: Metodologia utilizada nesta dissertação .....	20
Figura 2: GREat Tour: Guia de visitas móvel do laboratório GREat .....	23
Figura 3: GREat Mute: Aplicação em desenvolvimento no laboratório GREat .....	27
Figura 9: Partes componentes da norma SQuaRE .....	30
Figura 10: A metodologia GQM retirada de (ROCHA et al 2012) .....	33
Figura 4: A confiança e seus atributos adaptada de (SANTOS 2014) .....	37
Figura 5: A satisfação e seus atributos adaptada de (ISO 25010) .....	38
Figura 6: A confiança no funcionamento e seus atributos .....	39
Figura 7: A confiança no funcionamento dentro da ISO 25010, retirado de (ISO 25010) .....	40
Figura 8: Árvore da Confiança no Funcionamento adaptada de (AVIZIENIS et al 2004) .....	41
Figura 11: Árvore da Confiança Resultante da Fundamentação Teórica .....	43
Figura 12: Processo de identificação dos trabalhos relacionados .....	45
Figura 13: Relacionamento entre os trabalhos relacionados .....	49
Figura 14: Planejamento do GQM .....	56
Figura 15: Relação entre medidas e falhas .....	79
Figura 16: Modelo Final GQM .....	80
Figura 17: Planejamento para os estudos de caso .....	82
Figura 18: GREat Mute Versão 2.0 .....	86
Figura 19: GREat Mute Versão 1.0 .....	87
Figura 20: GREat Mute Versão 3.0 .....	88
Figura 21: Representação do funcionamento do sistema fAlert (PIVA et al 2014) .....	99
Figura 22: Interface gráfica do fAlert .....	100
Figura 23: Dispositivos espalhados pelo corpo do usuário .....	102
Figura 24: Casos de queda .....	103
Figura 25: (i) Resultados obtidos com o uso do acelerômetro para dispositivo atado ao peito .....	106
Figura 26: (ii) Resultados obtidos com o uso do acelerômetro e microfone para dispositivo atado ao peito .....	106
Figura 27: (iii) Resultados obtidos com o uso do acelerômetro, magnetômetro e microfone para dispositivo atado ao peito .....	106
Figura 28: Resultados obtidos para a medida precisão .....	107
Figura 29: Resultados obtidos para a medida sensibilidade .....	107
Figura 30: Resultados obtidos para a medida disponibilidade .....	108
Figura 31: Resultados obtidos para a medida especificidade .....	108

Figura 32: Resultados obtidos para a medida grau de desempenho.....	108
Figura 33: Resultados obtidos para a medida acurácia .....	109
Figura 34: Telas do GREat Tour .....	114

# Lista de Tabelas

Tabela 1: Normas ISO e seus propósitos .....	29
Tabela 2: Critérios Qualitativos para Definição de Medidas .....	31
Tabela 3: Template para a definição do objetivo de medição (Basili et al 2002).....	34
Tabela 4: Palavras-chave utilizadas para a busca de trabalhos .....	46
Tabela 5: Referência e descrição dos trabalhos relacionados .....	46
Tabela 6: Análise comparativa dos trabalhos relacionados.....	53
Tabela 7: Questões e desafios encontrados.....	54
Tabela 8: Abstract Sheet.....	58
Tabela 9: Medidas .....	61
Tabela 10: Perfil dos usuários envolvidos nos estudos de caso .....	83
Tabela 11: Dispositivos usados nos estudos de caso.....	83
Tabela 12: Configuração técnica do aparelho utilizado [PIVA et al 2014].....	99
Tabela 13: Informações providas pelos Sensores Utilizados para Detecção .....	102
Tabela 14: Quantidade de testes por categoria .....	104
Tabela 15: Medidas utilizadas para avaliar o fAlert.....	104
Tabela 16: Critérios qualitativos para avaliar medidas .....	119

# Lista de Abreviaturas

IHC	Interação Humano-Computador
ISO	<i>International Organisation for Standardisation</i>
GPS	<i>Global Positioning System</i>
GREat	Grupo de Redes de Computadores, Engenharia de Software, e Sistemas
QRCode	<i>Quick Responde Code</i>
GQM	<i>Goal-Question-Metric</i>

# Capítulo 1

## Introdução

Esta dissertação de mestrado tem como objetivo propor um conjunto de medidas de qualidade de *software* para avaliação da *Confiança no Funcionamento* em sistemas ubíquos para dispositivos móveis (e.g., *smartphones*, *tablets*, e *smart devices* em geral). Este capítulo introduz, na Seção 1.1, o contexto em que o trabalho de dissertação está inserido e a motivação para o seu desenvolvimento, na Seção 1.2 são descritos os objetivos da dissertação, na seção 1.3 é apresentada a metodologia utilizada nesta pesquisa e, por fim, a Seção 1.4 apresenta a estrutura organizacional deste trabalho.

### 1.1 Contextualização e Motivação

Os sistemas ubíquos possibilitam tornar real o conceito de computação em todo lugar e a qualquer momento, fazendo com que o uso dos recursos computacionais e a comunicação sejam transparentes para o usuário (YAU et al. 2002). Para tanto, é necessário que este tipo de sistema possua a habilidade de perceber o ambiente, as atividades e necessidades do usuário, sem que seja necessário determiná-los explicitamente (WEISER 1991) (SANTOS 2014) (SPÍNOLA 2007).

A autonomia desses sistemas pode ser proporcionada por meio do uso da sensibilidade ao contexto. Com o uso de informações contextuais, o usuário não precisa necessariamente realizar uma operação explícita (e.g., clicar em um botão) para que o sistema entenda isso como uma entrada. Muitas vezes, ações cotidianas (e.g., entrar em um ambiente) podem ser suficientes para que o sistema reaja provendo serviços (e.g., ligar o ar-condicionado de maneira automática) (DEY e ABOWD 2001) (LIMA 2011) (DEBASHIS e AMTAVA 2003).

Um exemplo de aplicação ubíqua que faz uso da sensibilidade ao contexto é um sistema de monitoramento de quedas em pessoas que necessitam de cuidados especiais, encontrado em (PIVA et al 2014). Utilizando dados obtidos por meio dos sensores do dispositivo móvel do usuário (e.g., localização, posição do dispositivo, variação de movimento), que podem servir para representar seu atual contexto, é possível detectar e alertar quedas.

Diante desse cenário, fazer com que aplicações percebam o contexto no qual o usuário está inserido pode ser fundamental para que a tomada de decisão seja a mais benéfica e adequada possível. Entretanto, a aceitação e, em consequência, a utilização em escala desse tipo de sistema necessitam, entre outras características, de confiança no seu funcionamento<sup>1</sup> (*dependability*) por parte do usuário. Neste caso, é necessário levar em consideração atributos de qualidade, tais como: disponibilidade (*availability*), confiabilidade (*reliability*), proteção (*safety*), confidencialidade (*confidentiality*), integridade (*integrity*) e manutenibilidade (*maintainability*). Esses atributos, segundo (AVIZIENIS et al 2004), podem impactar ou caracterizar a confiança no funcionamento, apresentada em mais detalhes no restante deste trabalho.

A obtenção das informações contextuais bem como seu uso por parte da aplicação acarreta em questões não triviais, associadas aos atributos de qualidade mencionados anteriormente e, conseqüentemente, relacionadas à confiança no funcionamento: (i) Como garantir que a informação contextual é verdadeira e realmente corresponde ao contexto atual do usuário? (ii) Como garantir que a informação contextual não pode ser acessada por quem não tem permissão? (iii) Como garantir que a informação contextual não foi comprometida?

Autores como (RANGANATHAN et al. 2005), (SCHOLTZ and CONSOLVO 2004), (SANTOS 2014) e (DAVIS et al. 2010) apresentam em seus trabalhos uma série de características que estão relacionadas às questões e aos atributos de qualidade apontados anteriormente. Por exemplo, a probabilidade de que o contexto tenha sido capturado corretamente e a porcentagem de erro do contexto inferido ou sentido são características diretamente relacionadas à questão (i), que implica em avaliar problemas relativos à confiabilidade, integridade, confidencialidade, proteção e segurança. A definição de quais informações o usuário deve disponibilizar e como estabelecer comunicações privadas são características associadas à questão (ii), o que acarreta em analisar problemas associados à confidencialidade, proteção e integridade. Além disso, de acordo com os autores e com o problema abordado em (iii), é importante garantir a acurácia e integridade das informações relacionadas ao contexto e as identidades das pessoas envolvidas.

---

<sup>1</sup> Nesta pesquisa de mestrado, o conceito *Dependability* é traduzido como *Confiança no Funcionamento*, baseado em discussões de pesquisadores da área de Engenharia de Software que tentaram chegar a um consenso sobre a tradução correta do termo para o português [<http://www.cs.kent.ac.uk/people/staff/rdl/CoF/>]

Por fim, é preciso também levar em consideração usuários que se movem nos limites físicos de um sistema ubíquo. Esses usuários podem trocar de dispositivos sem perder o acesso aos serviços, e o sistema deve garantir, assim, a sua disponibilidade.

Segundo (ROCHA et al 2010), os sistemas ubíquos impõem obstáculos ao uso das metodologias existentes na engenharia de software pois, devido as suas características específicas (e.g., mobilidade, heterogeneidade), não é possível projetá-los como um todo, de maneira coerente pois, alguns comportamentos do sistema podem ser conhecidos apenas no momento do uso. De acordo com (SANTOS 2014), existe uma necessidade de novas metodologias (e.g., modelos de qualidade, frameworks), bem como medidas de *software* diferenciadas, que reflitam essas características específicas dos sistemas ubíquos.

Uma das técnicas para a avaliação de características de qualidade é a realização de medições (SCHOLTZ e CONSOLVO 2004) e (SANTOS 2013). Uma medição consiste no processo contínuo de definição, coleta e análise de dados sobre o processo de desenvolvimento de software e seus produtos, fornecendo informações significativas com o objetivo de melhorá-los (ISO 25000). O resultado da medição é chamado de medida<sup>2</sup>. O conjunto de propriedades ou atributos de um produto de software, pelas quais a qualidade pode ser descrita e avaliada é chamado de característica. Definir medidas para a avaliação da característica confiança no funcionamento em sistemas ubíquos ainda é um desafio de pesquisa em aberto.

Apesar da confiança no funcionamento de um sistema ubíquo ser uma questão importante, por englobar diversos atributos de qualidade, garanti-la é uma tarefa difícil, devido as características e problemas relatados anteriormente. Por isso, pesquisadores tem reportado dificuldades para utilizar as formas de avaliação de qualidade existentes (e.g., modelos e medidas de qualidade) nesses sistemas (SANTOS 2014), buscando, como alternativa, adaptá-las, levando em consideração as diferenças (e.g., sensibilidade ao contexto, mobilidade e heterogeneidade) entre sistemas convencionais e ubíquos.

Logo, investigar e, eventualmente, adaptar e propor medidas que permitam a avaliação da confiança no funcionamento de sistemas ubíquos, em particular, aquelas aplicações para dispositivos móveis, é um problema desafiador.

---

<sup>2</sup> De acordo com a [ISO 25010], uma medida de qualidade é definida como uma variável a qual é atribuído um valor como resultado da medição [ISO 25010]. De acordo com a norma, as medidas quantitativas são também chamadas de métricas. Porém, neste trabalho, é utilizada apenas a nomenclatura medida.

## 1.2 Objetivos e Contribuições Esperadas

Considerando o que foi apresentado na seção anterior, este trabalho de dissertação de mestrado tem como objetivo investigar e, se necessário, adaptar e propor medidas de qualidade de *software* para avaliação da confiança no funcionamento (i.e., *dependability*) em sistemas ubíquos.

Para atingir esse objetivo, as seguintes metas foram definidas:

- (i) Busca na literatura por medidas para avaliação da confiança no funcionamento (e.g., *dependability*) em sistemas tradicionais e ubíquos;
- (ii) Adaptação de medidas existentes;
- (iii) Definição de novas medidas, utilizando o paradigma *Goal-Question-Metric* (GQM) para a definição das medidas;
- (iv) Aplicação das medidas em estudos de caso.

É importante ressaltar que os objetivos citados anteriormente estão inseridos em um projeto de pesquisa chamado Maximum<sup>3</sup> (Uma Abordagem baseada em Medições para Avaliação da Qualidade da Interação Humano-Computador (IHC) em Sistemas Ubíquos). Os resultados deste trabalho contribuem diretamente para este projeto e são uma evolução do trabalho encontrado em (SANTOS 2014), que propôs um conjunto de características e medidas para a avaliação da qualidade da IHC em sistemas ubíquos.

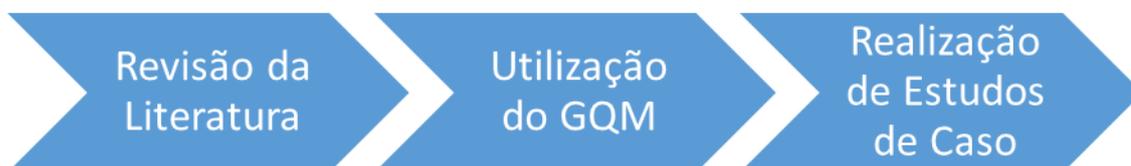
Espera-se, com este trabalho de pesquisa, também contribuir para a evolução da (ISO 25000) e que ele seja utilizado como apoio à atividade de avaliação da confiança no funcionamento em sistemas ubíquos com foco em aplicações rodando em dispositivos móveis. Com isso, pretende-se auxiliar o analista de qualidade na identificação de quais atributos estão impactando a confiança do usuário no funcionamento desse tipo de sistema e, em consequência, que os desenvolvedores usem os resultados dessas avaliações para melhorar a qualidade dessas aplicações, aumentando a aceitação e utilização das mesmas.

---

<sup>3</sup> Esse projeto foi financiado pela Fundação Cearense de Apoio Tecnológico e Pesquisas (FUNCAP) com o número de processo INC-0064-00012.01.00/12.

### 1.3 Metodologia Utilizada

Para alcançar o objetivo e as metas mencionados na seção anterior, a seguinte metodologia é utilizada nesta dissertação, conforme ilustrada na Figura 1.



**Figura 1: Metodologia utilizada nesta dissertação**

Para alcançar o objetivo (i), é necessária uma revisão da literatura. Com isso, deve ser possível identificar se existem medidas para a avaliação da confiança no funcionamento de sistemas ubíquos. Caso existam, é necessário verificar se elas levam em consideração apenas informações obtidas em tempo de projeto, pois, caso isto aconteça, é preciso uma adaptação para que sejam considerados também dados obtidos em tempo de execução.

Ainda sobre o processo de revisão da literatura, é fundamental destacar a importância do uso do mapeamento sistemático realizado por (SANTOS 2014), servindo este como base para esta pesquisa.

Além do uso ou, se for o caso, adaptação de medidas já existentes, é preciso também verificar a necessidade da definição de novas medidas. Com isso, para alcançar os objetivos (ii) e (iii) é utilizado o paradigma *Goal-Question-Metric* (GQM).

Apesar da existência de outros paradigmas igualmente eficazes, por exemplo, PSM (HUGHES 2000) e GQIM (PARK et al. 1996), o GQM foi escolhido pois é o mais adequado quando o objetivo de medição é claro e específico. Sendo este o caso desta pesquisa onde, o objetivo é avaliar a confiança no funcionamento de sistemas ubíquos para dispositivos móveis. Mais detalhes sobre o GQM, sua metodologia, seus objetivos de medição e sua aplicação nesta pesquisa podem ser encontrados nos Capítulos 2 e 4.

Por fim, para alcançar o objetivo (iv), são escolhidas três aplicações já existentes como estudos de caso. Estudos de caso permitem a coleta de dados quantitativos e qualitativos. Além disso, podem ser conduzidos em configurações do mundo real, possibilitando, portanto, um maior grau de realismo (YIN 1994). A análise dos resultados obtidos com a avaliação dos três estudos de caso permite avaliar se as

medidas de qualidade de *software* definidas neste trabalho são capazes de fornecer informações significativas sobre a característica de qualidade confiança no funcionamento, tendo como foco os sistemas ubíquos.

## 1.4 Organização da dissertação

Além deste capítulo introdutório, esta dissertação está estruturada em mais cinco capítulos e 4 apêndices.

**Capítulo 2 – Referencial Teórico:** tem por objetivo fornecer o embasamento teórico acerca das áreas que compõem esse trabalho, que são Sistemas Ubíquos, Qualidade de Software e Confiança no Funcionamento em Sistemas de Software.

**Capítulo 3 – Trabalhos Relacionados:** apresenta a metodologia utilizada para a busca de trabalhos na área, bem como os resultados encontrados, comparando ainda esses resultados com o trabalho desenvolvido nesta dissertação.

**Capítulo 4 – Medidas de Qualidade de *Software* para Avaliação da Confiança no Funcionamento em Sistemas Ubíquos:** apresenta as medidas de qualidade de software definidas para avaliação da confiança no funcionamento de sistemas ubíquos. Também apresenta a metodologia utilizada para a definição dessas medidas.

**Capítulo 5 – Estudo de Caso:** apresenta os resultados obtidos nos estudos de caso realizados para aplicar as medidas propostas. E os benefícios alcançados com esses resultados.

**Capítulo 6 – Conclusão:** apresenta um resumo dos resultados alcançados neste trabalho e seus direcionamentos futuros.

**Apêndice A – Guia de Medidas:** guia onde as medidas definidas no Capítulo 4 são apresentadas com mais detalhes.

**Apêndice B – Formulário para traçar o perfil do usuário:** formulário utilizado com as pessoas que participam dos testes relacionados aos estudos de caso apresentados no Capítulo 5.

**Apêndice C – Questionário para ser utilizado pelo usuário:** questionário a ser utilizado como auxílio para o método de coleta de algumas medidas apresentadas no apêndice A, onde o ponto de vista a ser avaliado é o do usuário final.

**Apêndice D – Questionário para ser utilizado pelo desenvolvedor:** questionário a ser utilizado como auxílio para o método de coleta de algumas

medidas apresentadas no apêndice A, onde o ponto de vista a ser avaliado é o do desenvolvedor.

# Capítulo 2

## Referencial Teórico

Neste capítulo é apresentada uma revisão bibliográfica das áreas de estudo relacionadas a esta pesquisa, a qual busca facilitar o entendimento dos desafios e soluções apontados neste trabalho. A Seção 2.1 apresenta definições e princípios da área de Sistemas Ubíquos. A Seção 2.2 apresenta as principais definições da área de Qualidade de Software. e, finalmente, a Seção 2.3 apresenta os conceitos da área de Confiança, em especial a Confiança no Funcionamento de Sistemas.

### 2.1 Sistemas Ubíquos

Imagine-se entrando em um laboratório de pesquisa que você ainda não conhece, acompanhado somente de um dispositivo móvel onde está instalada uma aplicação que servirá como um guia de visitas do local, como o exemplo ilustrado na Figura 1.



Figura 2: GREAt Tour: <sup>4</sup> Guia de visitas móvel do laboratório GREAt

---

<sup>4</sup> Guia de visitas móveis sensível ao contexto, produto de uma Linha de Produto de Software Sensível ao Contexto chamada Moline, desenvolvida pelo grupo GREAt e que pode ser encontrada em [Marinho et al. 2012]

Ao entrar, você é imediatamente notificado sobre a sala na qual se encontra, acompanhado de um mapa geral que retrata os demais ambientes. Sendo guiado pelas informações disponibilizadas pelo aplicativo você é capaz de conhecer todo o laboratório, obtendo informações acerca de funcionários, pesquisadores, projetos de pesquisa desenvolvidos e trabalhos relacionados. À medida que se movimenta pelo ambiente, você é atualizado de informações dependendo de sua posição, sem a necessidade de realizar operações diretas com o dispositivo (e.g, clicar em um botão).

Portanto, munido apenas de um dispositivo móvel você é capaz de conhecer e passear por um local sobre o qual você não tem nenhum conhecimento. Esse é um típico exemplo de ambiente que remete à ideia de computação ubíqua, idealizada em 1991, por Marc Weiser.

Considerado o pai da computação ubíqua, Weiser afirmou que no futuro computadores irão habitar os mais triviais objetos: etiquetas de roupas, xícaras de café, interruptores de luz, canetas, entre outros, de forma invisível para o usuário. Com isso, as pessoas devem aprender a conviver com computadores, e não apenas interagir com eles (WEISER 1991).

No mundo idealizado pelo autor, a computação move-se para fora das estações de trabalho e computadores pessoais, tornando-se indistinguível em nossa vida cotidiana. Nesta concepção, o computador tem a capacidade de obter informações do ambiente no qual está e utilizá-las para prover serviços de maneira natural. Assim, surge a capacidade de computadores agirem de forma autônoma e “inteligente” no ambiente.

A fim de se obter uma definição mais precisa e completa sobre computação ubíqua, outras definições podem ser encontradas na literatura. Alguns exemplos são:

- “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” (SPINOLA et al 2007).

Apesar de definidas em diferentes contextos de pesquisa, as diversas definições encontradas para sistemas ubíquos concordam com relação às ideias-chave que podem ser extraídas do trabalho de (WEISER 1991): invisibilidade, transparência, naturalidade e, conseqüentemente, autonomia.

É possível, então, estabelecer um conjunto de princípios associados aos sistemas ubíquos e que se fazem pertinentes a este trabalho de pesquisa (LIMA 2011):

- Onipresença dos serviços: usuários podem se mover nos limites físicos de um sistema ubíquo e trocar de dispositivos sem perder o acesso aos serviços;
- Captura de experiências e intenções: o sistema deve ser capaz de perceber o ambiente, as atividades e necessidades do usuário, sem que seja necessário determiná-las explicitamente;
- Heterogeneidade de serviços e dispositivos - habilidade de possuir uma grande variedade de dispositivos e serviços;
- Interoperabilidade - capacidade de diferentes tipos de dispositivos e serviços interagirem automaticamente sem a intervenção do usuário;
- 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;
- 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;
- 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; e
- Adaptação ao comportamento - habilidade de se adaptar ao contexto e prover serviços e/ou informações relevantes para os usuários.

Diante desse contexto, neste trabalho de pesquisa, os sistemas ubíquos considerados são, por natureza, sensíveis ao contexto.

A computação sensível ao contexto aborda a ideia de que os computadores podem tanto perceber quanto reagir ao ambiente no qual estão inseridos buscando facilitar as atividades humanas. Os dispositivos usam informações sobre as circunstâncias em que se encontram e, baseados em regras ou estímulos inteligentes, reagem de acordo (DEBASHIS E AMITAVA 2003). Segundo (LIMA 2011) é impossível conceber um sistema ubíquo que

não seja sensível ao contexto, já que os princípios particulares desses sistemas, mencionados anteriormente, precisam da sensibilidade ao contexto para serem alcançados.

Uma das principais definições de contexto é apresentada por (DEY 2001). Segundo o autor, 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 considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a aplicação.

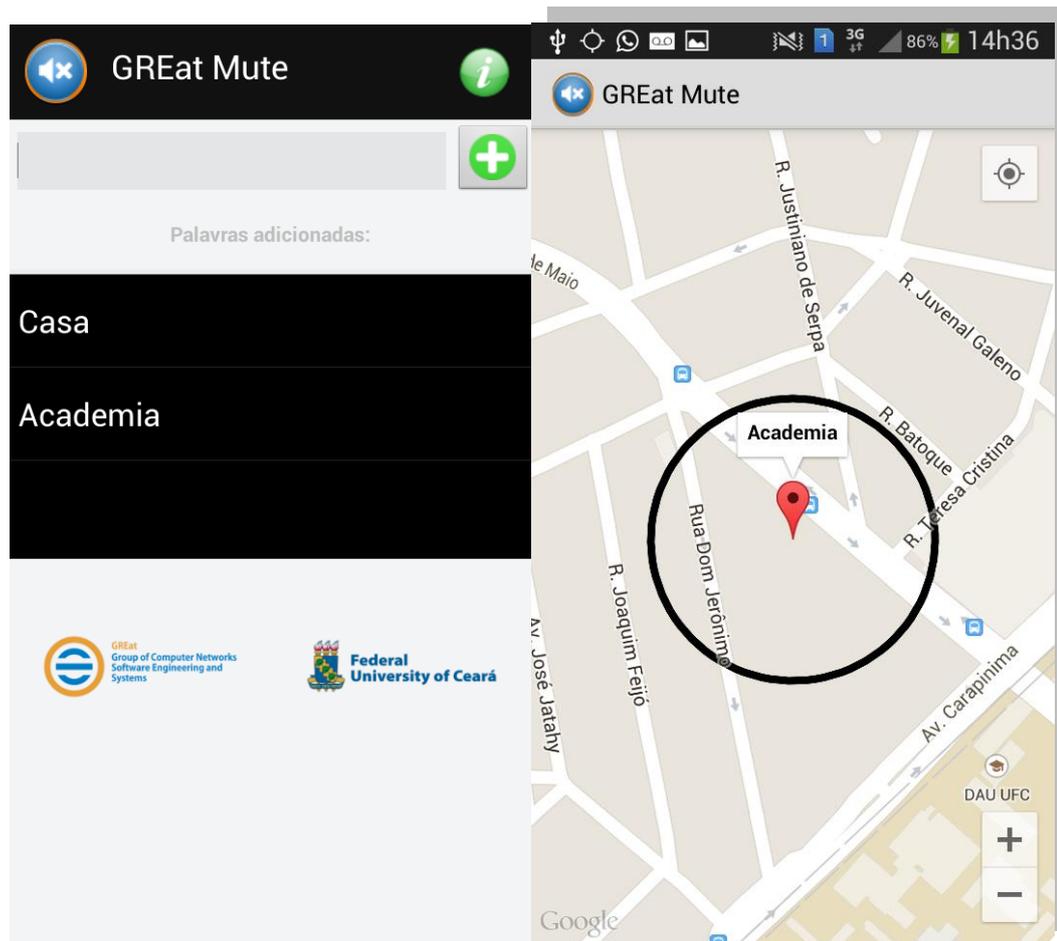
Segundo (DEY 2001), os aspectos relacionados à obtenção das informações contextuais são descritos com base em cinco dimensões, também conhecidas como “cinco W’s H”, definidos a seguir:

- Quem (*Who*) - relaciona-se a identidade do usuário;
- Onde (*Where*) - trata-se da localização do usuário;
- O que (*What*) - foca nas atividades do usuário;
- Quando (*When*) - relacionado ao contexto temporal e
- Por que (*How*) - está relacionado ao contexto que levou o usuário a fazer determinada ação.

De acordo com o autor, um sistema é dito sensível ao contexto se utiliza informação contextual para disponibilizar informações ou serviços relevantes para o usuário.

Para melhor compreensão da relevância de um serviço que esse sistema pode prover, suponha agora o seguinte cenário: imagine que você está participando de uma importante reunião e opta por colocar o seu dispositivo móvel no modo silencioso, para evitar interrupções. Ao sair da reunião, você esquece de ativar o modo normal do dispositivo e perde uma série de ligações importantes porque não ouviu as chamadas.

Esse inconveniente poderia ser facilmente evitado com o uso de uma aplicação ubíqua para dispositivos móveis que, utilizando informações sobre compromissos cadastrados na agenda pessoal do usuário e sua atual localização, fosse capaz de ativar e desativar o modo silencioso em momentos considerados adequados. Um exemplo de aplicação desse tipo é ilustrado na Figura 2.



**Figura 3: GREAt Mute: Aplicação em desenvolvimento no laboratório GREAt**

Nas três aplicações apresentadas até então (sistema de detecção de quedas, Capítulo 1, guia de visitas e aplicação para automatizar o uso do modo silencioso, Capítulo 2) é possível perceber que a aceitação e utilização desse tipo sistema requer um certo nível de confiança por parte do usuário. Delegar o monitoramento de pessoas que necessitam de cuidados especiais à um sistema de software, passear em um local ao qual nunca foi baseado somente nas informações disponibilizadas por uma aplicação e delegar as configurações do seu aparelho móvel a um sistema são ações baseadas na confiança no funcionamento do sistema<sup>5</sup>.

Segundo (SANTOS 2014), baseado em um mapeamento sistemático realizado pelo autor, diversas características de qualidade (e.g., usabilidade, transparência, disponibilidade, confiança) exercem forte influência sobre a aceitação e utilização de sistemas ubíquos por parte do usuário. Ainda segundo o autor, uma das possíveis técnicas para avaliar essas características e, porventura, impulsionar o uso real de sistemas ubíquos, é a realização de

<sup>5</sup> Neste trabalho, ao falar sobre sistema, estão sendo considerados apenas sistemas ubíquos para dispositivos móveis (e.g. smartphones, tablets). Dando ênfase ao seu comportamento exterior (e.g., comportamento percebido pelo usuário final).

medições. Para tanto é necessário o entendimento de diversos conceitos da área de qualidade de software.

## 2.2 Qualidade de Software

Ao longo dos anos diversas definições de qualidade de software foram desenvolvidas. Para (PETERS 2002) a qualidade de software é avaliada em termos de atributos de alto nível chamados fatores, que são medidos em relação a atributos de baixo nível chamados de critérios. Segundo (SANDERS 1994) um produto de software apresenta qualidade dependendo do grau de satisfação das necessidades dos clientes sob todos os aspectos do produto. Para (PRESSMAN 2009) a qualidade de software é a conformidade com os requisitos funcionais e de desempenho que foram explicitamente declarados e a características implícitas que são esperadas de todo software desenvolvido por profissionais.

Diante dessa diversidade na definição da palavra qualidade, (PRESSMAN 2009) sugere que a qualidade de software seja implementada e não somente uma idéia ou desejo que uma organização venha a ter. Para tanto, o autor faz as seguintes colocações sobre qualidade de software:

- I. Definir explicitamente o termo qualidade de software, quando o mesmo é dito;
- II. Criar um conjunto de atividades que irão ajudar a garantir que cada produto de trabalho da engenharia de software exiba alta qualidade;
- III. Realizar atividades da qualidade em cada projeto de software;
- IV. Usar medidas para desenvolver estratégias para a melhoria do produto e processo de software;

Seguindo a recomendação (I), mencionada anteriormente, nesta dissertação a qualidade é vista como o grau no qual um produto de software satisfaz as necessidades explícitas e implícitas quando usado sob condições específicas (ISO 25000). Essa definição é adotada pois, a norma ISO 25000 leva em consideração o contexto de uso ao avaliar algumas de suas características de qualidade. Estando este critério intimamente relacionado aos objetivos e motivação desta pesquisa.

Diante desse cenário, algumas das principais questões a serem analisadas quando o assunto é qualidade de software é como garanti-la e avaliá-la. Para tanto, surgiram as normas de qualidade<sup>6</sup>

## 2.2.1 Normas de Qualidade

Ao longo dos anos, as normas ISO<sup>7</sup> tornaram-se sinônimo de preocupação com qualidade. As suas normas oferecem uma base comum para julgar a qualidade de produtos e definir critérios em contratos e negociações.

Existe um número expressivo de normas técnicas internacionais relacionadas com software. Em sua maioria, as normas de software têm um caráter mais informativo do que regulatório. Na Tabela 1, adaptada de (KOSCIANSKI 2007) é possível ver algumas das normas existentes bem como um resumo do seu propósito.

**Tabela 1: Normas ISO e seus propósitos**

<i>Norma</i>	<i>Propósito</i>
ISO 12207	Processos de ciclo de vida de software
ISO/IEC 12119:1994	Pacotes de Software – Requisitos de qualidade e testes
ISO/IEC 14598-1:1999	Avaliação de Qualidade de produtos de software
ISO/IEC 9126-1:2001	Modelo de Qualidade – Características
ISO/IEC 25000:2005	Modelo de Qualidade de software, nova versão das séries 14.598 e 9.126
ISO 9241:1998	Ergonomia de software
ISO/IEC 20926:2003	Medida de software por pontos de função

<sup>6</sup> *Uma norma de qualidade nasce da necessidade de padronização, necessidade esta que pode ser identificada na indústria ou no meio acadêmico.*

<sup>7</sup> *A ISO é uma família de normas e tem caráter genérico, servindo de base para qualquer organização em qualquer ramo de atividade, que queira realizar o controle de qualidade dos produtos ou serviços oferecidos (Koscianski 2007).*

ISO/IEC 90000-3:2004	Diretivas para aplicação da ISSO 9001 ao software
ISO 9001:2000	Requisitos para sistemas de gerenciamento de qualidade (aplicável a qualquer empresa, de software ou não)

Conforme mencionado no início desta seção, a definição de qualidade adotada nesta pesquisa é proposta pela ISO 25000, também conhecida como norma ou projeto SQuaRE. A norma ISO/IEC 25000 é uma evolução das normas ISO/IEC 9126 e ISO/IEC 14598 que tratam de qualidade de produtos de software.

Segundo (SANTOS 2014), o projeto SQuaRE, ilustrado na Figura 4, reorganizou o conjunto dos 10 componentes que formam as normas ISO/IEC 9126 e ISO/IEC 14598, mas não realizou mudanças radicais no material preexistente. O modelo hierárquico de qualidade proposto na 9126 continuou válido, assim como diversos aspectos organizacionais propostos pela 14598.

	Modelo de Qualidade 2501n	Avaliação 2504n
Requisitos de Qualidade 2503n	Gerenciamento de Qualidade 2501n	
	Medições 2501n	

**Figura 4: Partes componentes da norma SQuaRE**

Como podemos observar na Figura 9, cada divisão é composta por um conjunto de documentos, referenciados da forma 250Xn (onde x varia entre 1 e 4) e trata de um assunto em particular, definidos de maneira sucinta, 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.

É importante ressaltar também que a norma SQuaRE identifica três tipos de qualidade, que correspondem a diferentes situações envolvendo um produto de software (REF#):

- Qualidade Interna: é avaliada na representação estática do produto e, é exatamente a arquitetura interna que é levada em conta. A qualidade de organização do código ou a complexidade algorítmica são exemplos de critérios que podem indicar, respectivamente, prováveis custos de manutenção e provável velocidade de execução.
- Qualidade Externa: considera o produto como uma caixa-preta. A realização de testes corresponde a verificar a qualidade externa. A qualidade externa é uma estimativa da qualidade em uso; ambas podem ser muito próximas, mas não são equivalentes.
- Qualidade em Uso: corresponde ao ponto de vista de um usuário. A qualidade em uso, quando se refere a um programa sendo executado, depende de diversos fatores como o hardware utilizado, o treinamento do usuário, as condições ambientes, as tarefas realizadas e etc.

## 2.2.2 Medidas de Qualidade

Segundo a (ISO 25000), uma medida de qualidade de software é uma variável ao qual é atribuído um valor, como o resultado da medição. De acordo com (KOSCIANSKI E SOARES 2007), dentre as dificuldades para se definir medidas de qualidade de software, pode-se citar duas: a variedade de aspectos a considerar e a presença de muitos elementos intangíveis.

Para tanto, segundo (SCALET 1999), alguns critérios devem ser levados em consideração pelo avaliador ao elaborar as medidas, estes critérios podem ser vistos na Tabela 2, adaptada de (KOSCIANSKI E SOARES 2007).

**Tabela 2: Critérios Qualitativos para Definição de Medidas**

<i>Significância</i>	<i>Os resultados obtidos devem agregar informação útil a avaliação da qualidade.</i>
----------------------	--

Custo e Complexidade de Aplicação	O custo e a complexidade de uma medida deve ser compatível com a avaliação a ser realizada.
Repetibilidade	As medidas devem ser repetíveis, exceto algumas que dificilmente podem ser repetidas por envolverem condições não controláveis.
Reprodutabilidade	Os resultados de uma avaliação devem ser os mesmos para avaliadores diferentes.
Objetividade	A opinião de pessoas envolvidas devem ser limitadas ou totalmente evitadas.
Imparcialidade	As medidas devem ser imparciais, e deve se ter em mente que uma escolha não criteriosa de parâmetros pode levar a resultados diferentes e favorecer ou não um produto.
Evidência para Validação	As medidas devem prover evidências para sua validação. Por exemplo, se o resultado de uma avaliação é uma curva de tendência, é preciso prover informação a respeito do grau de confiabilidade dos valores apresentados.

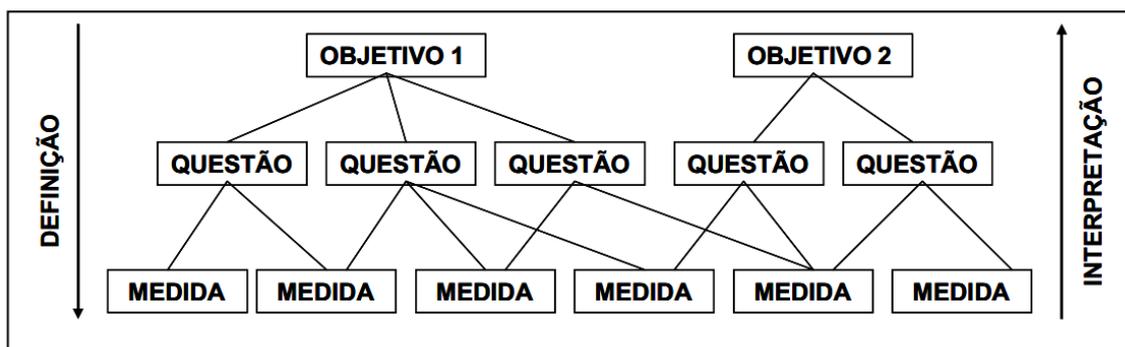
De acordo com (ROCHA et al 2012), ao definir medidas de qualidade de *software* é necessário também ter clareza sobre a adequação e utilidade destas, para que seja possível garantir que as informações obtidas com a sua coleta apoiem a tomada de decisão, respeitando os objetivos definidos para o sistema que está sendo avaliado. Para tanto, segundo (SANTOS 2014), é possível utilizar diferentes metodologias para a definição de medidas (e.g., GQM, GQiM, PSM).

O *Goal-Question-Metric* (GQM) 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 et al. 2002).

Devido à dificuldade em se derivar objetivos de medição no GQM, surgiu, como auxílio, a abordagem *Goal-Question-Indicator-Metric* (GQiM). A metodologia é composta por dez passos e procura alcançar os objetivos de negócio preparando um plano de medições. O *Practical Software Measurement* (PSM), por sua vez, é uma abordagem para especificar

formalmente medidas de qualidade de *software* e como aplicá-las em um processo de medição (HUGHES, 2000).

Nesta dissertação de mestrado, a metodologia utilizada para a definição de medidas é o GQM. Essa metodologia é bastante difundida e utilizada devido à sua simplicidade. Segundo (ROCHA et al 2012), por ter uma estrutura *top-down*, utilizando o GQM é possível identificar as medidas a partir de questões associadas ao objetivo de medição, como ilustrado na Figura 10.



**Figura 5: A metodologia GQM retirada de (ROCHA et al 2012)**

Como se pode observar na Figura 10, o resultado da aplicação do GQM é uma estrutura composta de três níveis. Segundo (ROCHA et al 2012), no nível 1, ou nível conceitual, estão os objetivos, no nível 2, ou operacional, estão as questões e, no nível 3, ou quantitativo, encontram-se as medidas.

Ainda segundo o autor, com base em (SOLLINGEN e BERGOUTH 1999), após a definição dos objetivos, questões e medidas é necessário produzir um plano de medição e análise. Para isso, é fundamental definir, entre outros:

- Possíveis valores para as medidas;
- Métodos para a coleta das medidas;
- Responsáveis para a coleta das medidas;
- O momento de coleta; e
- Procedimentos para a análise das medidas.

Como mencionado no Capítulo 1, o GQM é a técnica mais adequada quando o objetivo da medição é claro e específico, como é o caso deste trabalho, onde o objetivo é avaliar a confiança no funcionamento de sistemas ubíquos que executam em dispositivos móveis.

Por esse motivo, essa metodologia foi a escolhida e, é apresentada com mais detalhes a seguir.

### 2.2.3 *Goal-Question-Metric (GQM)*

Segundo (BASILI 2002), a metodologia GQM é dividida nas seguintes etapas: planejamento, definição, coleta de dados e interpretação de resultados, apresentadas em detalhes a seguir.

Na etapa de planejamento é necessário definir a equipe que participará do GQM, a área de melhoria e os projetos que farão parte da aplicação do método. O principal objetivo dessa fase é coletar todas as informações necessárias para uma avaliação de sucesso. Para isso, como mencionado anteriormente, é necessário estabelecer um time GQM, uma área de melhoria, um projeto de aplicação e o time do projeto.

Na fase de definição é preciso elucidar os objetivos do GQM, as questões a serem respondidas, além de definir e refinar as medidas.

A fase de definição de medidas é composta por:

- Definição do objetivo de medição, com objeto, propósito e foco;
- Entrevistas GQM com especialistas;
- Utilização dos resultados das entrevistas para derivar questões; e, por fim,
- Medidas.

Cada um dos componentes é descrito em detalhes a seguir:

Para a definição do objetivo de medição é utilizado o *template* definido por (BASILLI et al. 2002), apresentado na Tabela 3. De acordo com esse *template* devem ser definidos os objetos sob medição, os propósitos do objetivo, os focos de qualidade e o ponto de vista (quem mede o objeto).

**Tabela 3: Template para a definição do objetivo de medição (Basili et al 2002)**

<i>Analisar</i>	<i>Os sistemas ubíquos que executam em dispositivos móveis</i>
Com o propósito de	Avaliar
Com respeito à	Confiança no Funcionamento

Sob o ponto de vista do	Desenvolvedor e usuário
-------------------------	-------------------------

No caso deste trabalho de pesquisa, o objeto sob medição é o sistema ubíquo para dispositivo móvel, o propósito é avaliar e os focos de qualidade são os cinco atributos de qualidade relacionados à confiança no funcionamento: Disponibilidade, Integridade, Confiabilidade, Manutenibilidade e Proteção. Por fim, o ponto de vista é o do usuário e o do desenvolvedor.

Segundo (MARTINS 2011), o objetivo de medição é definido em um nível abstrato, por isso, as questões são o refinamento do objetivo em um nível mais operacional e mais simples de interpretar. Logo, as questões devem ser definidas em um nível intermediário de abstração entre as medidas e o objetivo, para facilitar a interpretação dos dados coletados.

Após a definição do objetivo de medição, é necessário realizar as entrevistas GQM com o time estabelecido na fase de planejamento. Com isso, será possível identificar quais questões são relevantes com respeito ao objetivo e, quais as possíveis medidas capazes de responde-las.

Para guiar as entrevistas GQM alguns tópicos devem ser levados em consideração, são eles (GRESSE et al 1995) (RUHE et al 1994):

- **Foco de Qualidade:** para o entrevistado, o que significa o(s)foco(s) de qualidade especificado(s) no objetivo de medição do GQM? Por exemplo, se o foco de qualidade definido foi *Confiabilidade*, então o que significa *Confiabilidade* para o entrevistado?
- **Hipótese de *baseline*:** Para cada foco de qualidade declarado: quais as possíveis medidas para avalia-lo? E quais os valores esperados para cada medida?
- **Fatores de Variação:** Quais fatores específicos do tipo de sistema avaliado têm possivelmente um impacto nos focos de qualidade? Seguindo a linha dos exemplos anteriores, para o entrevistado, sensibilidade ao contexto pode tem impacto sobre a *Confiabilidade*?

- **Impacto na Hipótese de *baseline*:** Para cada fator de variação declarado: para o entrevistado, qual é a estimativa do impacto do fator de variação no foco de qualidade?

Por fim, nas fases de coleta e interpretação é necessário coletar os dados, com base nas medidas definidas e, interpretá-los buscando responder as questões estabelecidas.

A fase de coleta consiste em coletar, armazenar e processar todas as informações necessárias para calcular as medidas. Definindo, portanto, como será a obtenção dos dados que serão utilizados na medição. De acordo com (VAN SOLINGEN E BERGHOUT, 1999), durante a fase de coleta de dados, é importante observar quatro 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?

Em relação a interpretação, o objetivo nesse estágio é analisar os resultados obtidos na coleta dos dados e utilizá-los para interpretar as questões e os objetivos de medição definidos.

## 2.3 Confiança no Funcionamento de Sistemas de Software

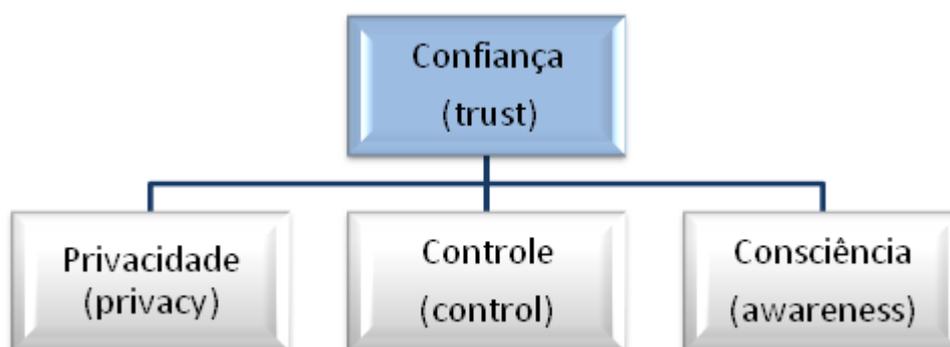
Apesar do trabalho realizado por (SANTOS 2014), as medidas do autor não são aplicáveis a todas as características levantadas durante o seu mapeamento sistemático. Destacando assim, a possibilidade e necessidade da definição de medidas para outras características, entre elas, a confiança (traduzida do inglês *trust*).

Diferentemente de características como disponibilidade, que possui uma definição mais clara e precisa na literatura (e.g., prover serviço sempre que requisitado (AVIZIENIS et al 2004) (STALLINGS 2012)), o termo confiança traz uma certa subjetividade em sua definição quando usado em diferentes áreas.

Para que as medidas de qualidade relacionadas a este trabalho possam realmente avaliar da maneira mais precisa possível essa característica, é necessário estabelecer de maneira objetiva e direta o que definimos como confiança. Para tanto, é necessário avaliar e respeitar uma série de conceitos estabelecidos nas áreas de qualidade de software,

engenharia de software e redes de computadores que possibilitam estabelecer e guiar quais aspectos podem ser considerados como atributos da confiança do usuário em um sistema.

Segundo (SANTOS 2014) e (SCHOLTZ E CONSOLVO 2004), a confiança (traduzida do inglês *trust* em ambos os trabalhos), apresentada na Figura 4, pode ser vista como a crença do usuário que o sistema utilizará os seus dados pessoais corretamente sem causar nenhum prejuízo ou dano. Para tanto, os autores apontam como atributos importantes para a garantia dessa confiança, aspectos relativos à privacidade (traduzida do inglês *privacy*, representa o grau de sigilo das informações pessoais do usuário, seja para outros usuários do sistema ou terceiros), consciência (traduzida do inglês *awareness*, representa a capacidade do usuário de entender quais das suas informações são ser utilizadas pelo sistema e quando) e controle (traduzida do inglês *control*, representa a capacidade do usuário de controlar quais das suas informações podem ser utilizadas pelo sistema e quando podem).



**Figura 6: A confiança e seus atributos adaptada de (SANTOS 2014)**

Já para a norma ISO 25010, o termo confiança (traduzida do inglês *trust*), apresentada na Figura 5, representa o grau de confiança que um usuário ou qualquer outra parte interessada tem de que um produto ou sistema irá se comportar conforme o esperado. Ainda segundo a norma, a confiança deve ser vista como uma subcaracterística da característica satisfação (traduzida do inglês *satisfaction*) que representa, por sua vez, o grau em que as necessidades dos usuários são satisfeitas, quando um produto ou sistema é utilizado em um contexto de uso específico.

Além da confiança, compõem o grau de satisfação as subcaracterísticas conforto (traduzida do inglês *comfort*, representa o grau com o qual o usuário está satisfeito com o conforto físico), utilidade (traduzida do inglês *usefulness*, representa o grau de satisfação do usuário ao perceber que o sistema consegue perceber os seus objetivos) e prazer (traduzida

do inglês *pleasure*, representa o grau em que um usuário obtém prazer ao cumprir as suas necessidades pessoais).

Diante desse cenário, o nosso primeiro passo é estabelecer a confiança como uma característica de qualidade, assim como satisfação em (ISO 25010) e, após isso, refiná-la em uma série de subcaracterísticas.



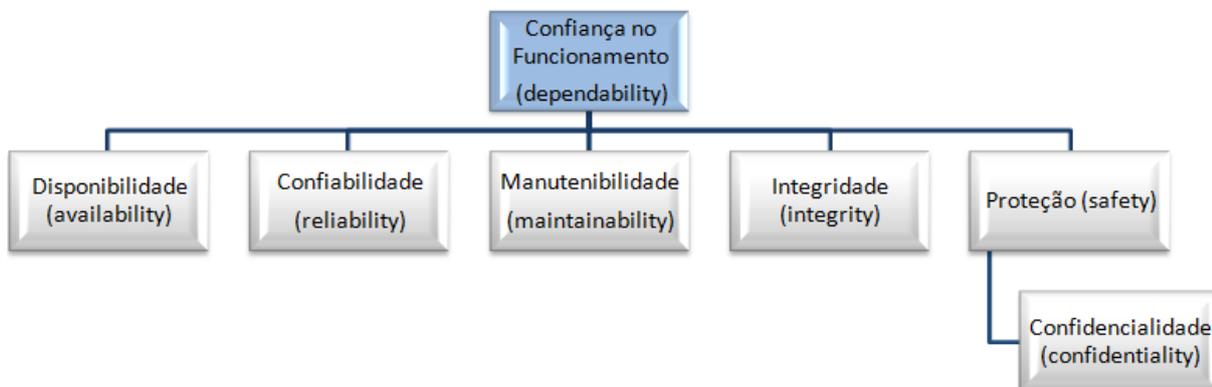
**Figura 7: A satisfação e seus atributos adaptada de (ISO 25010)**

Sendo assim, o restante desse texto é baseado no conceito de confiança no funcionamento (traduzido do inglês *dependability*), definido por (AVIZIENIS et al 2004). Segundo o autor, o termo é proposto como representação dos esforços das comunidades de confiança no funcionamento e segurança (traduzida do inglês *security*) buscando unificar vários aspectos dessas duas áreas, servindo como uma espécie de guia de termos e conceitos. Assim, diante da precisão e aceitabilidade por parte da comunidade científica do termo estabelecido, é possível definir medidas de qualidade de software que avaliem a confiança do usuário no funcionamento do sistema, levando em consideração um conjunto de atributos mais amplo do que o proposto por (SANTOS 2014) e (SCHOLTZ e CONSOLVO 2004) e com uma maior granularidade do que o proposto pela (ISO 25010).

É importante destacar também que não faz parte das intenções deste trabalho discutir sobre as controvérsias geradas por parte da literatura (e.g., confiança no funcionamento (*dependability*) é sinônimo de confiança (*trust*) (FOGG e TSENG 1999) (GOILAU 2003) (SOLNER 2012)) mas sim, fazer uso do que foi estabelecido por (AVIZIENIS et al 2004) em benefício da área de qualidade de software e medições.

Assim sendo, a confiança avaliada com as medidas relativas a este trabalho representa a capacidade do usuário de poder justificadamente confiar no funcionamento de um sistema. Para tanto, essa confiança é justificada com base na confiabilidade (traduzida

do inglês *reliability*), disponibilidade (traduzida do inglês *availability*), integridade (traduzida do inglês *integrity*), manutenibilidade (traduzida do inglês *maintainability*), proteção (traduzida do inglês *safety*) e confidencialidade (traduzida do inglês *confidentiality*) do mesmo.



**Figura 8: A confiança no funcionamento e seus atributos**

Segundo (AVIZIENIS et al 2004), a confiabilidade representa a habilidade de prestar serviço conforme o esperado de maneira contínua, a disponibilidade implica em prestar serviço correto sempre que requisitado, a integridade implica na ausência de alterações impróprias para o sistema, a manutenibilidade representa a habilidade de sofrer modificações e reparos, a proteção implica na ausência de consequências catastróficas para o usuário e o ambiente e, por fim, a confidencialidade implica na ausência de divulgação não autorizada de informação. Ainda segundo o autor é importante observar que a segurança (do inglês *security*) e a confiança no funcionamento possuem diversos atributos em comum.

Segundo (ROCHA 2010): “esses atributos podem ser tratados em conjunto desde que disponibilidade, confiabilidade e integridade sejam considerados simultaneamente, a disponibilidade seja apenas para os usuários autorizados e que na definição de integridade o termo impróprio seja entendido como não autorizado”.

De acordo com a norma ISO 25010, os atributos da confiança no funcionamento estão mapeados em seu modelo de características como apresentado na Figura 7:

Clause	ISO/IEC 25010	Dependability
4.1.1	Effectiveness	*
4.1.2	Efficiency	*
4.1.3	Satisfaction	*
4.1.4	Freedom from risk	Safety
4.1.5	Context coverage	*
4.2.1	Functional suitability	*
4.2.2	Performance efficiency	*
4.2.3	Compatibility	*
4.2.4	Usability	*
4.2.5	Reliability	Reliability
4.2.5.2	Availability	Availability
4.2.6	Security	
4.2.6.1	Confidentiality	Confidentiality
4.2.6.2	Integrity	Integrity
4.2.7	Maintainability	Maintainability
4.2.8	Portability	*

**Figura 9: A confiança no funcionamento dentro da ISO 25010, retirado de (ISO 25010)**

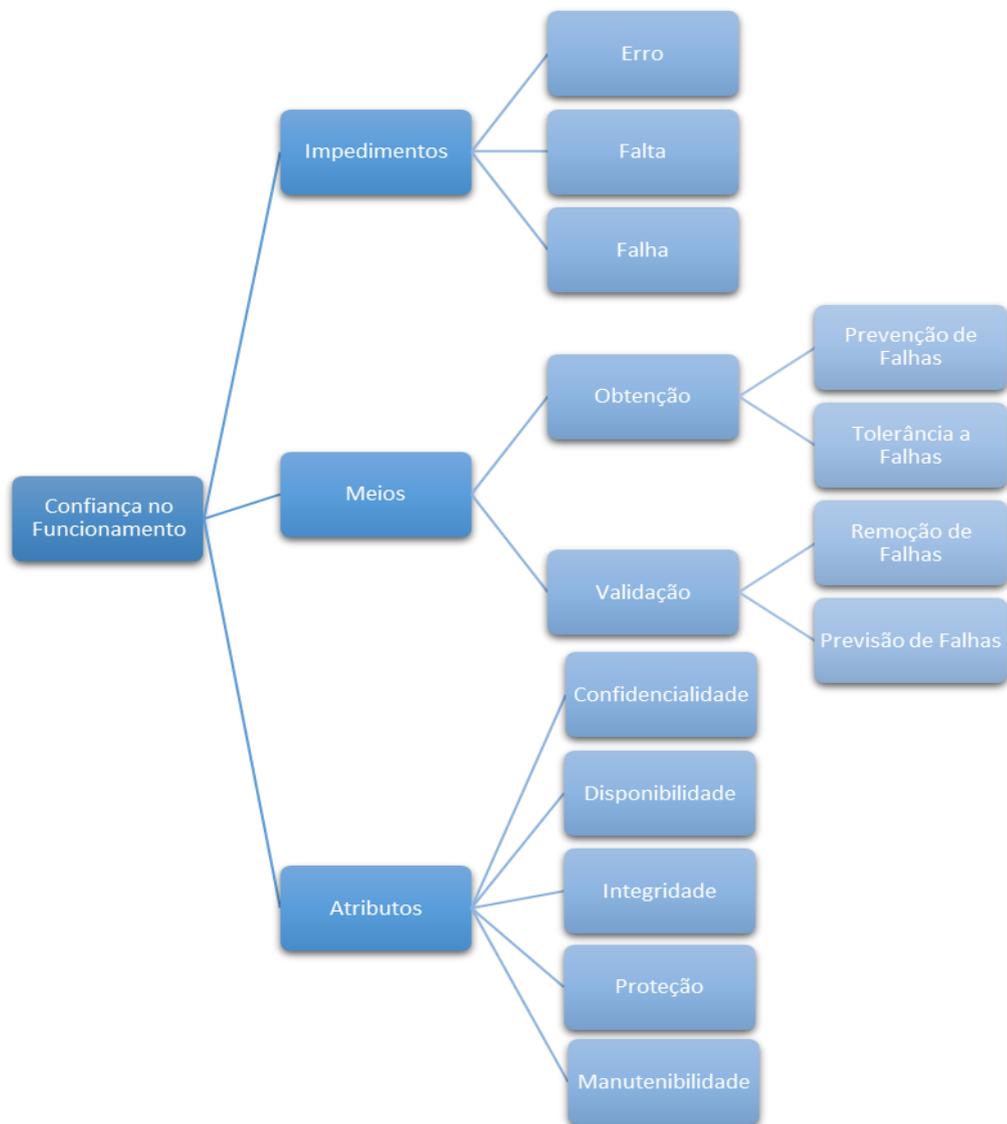
Neste trabalho, o agrupamento dos atributos mencionados por (AVIZIENIS et al 2004) é feito de modo que todos eles estejam associados, como subcaracterísticas, a mesma característica: a confiança no funcionamento. Com isso, levando em consideração o domínio desta pesquisa (sistemas ubíquos), aspectos como a proteção e confiabilidade do sistema passam a ser avaliados levando em consideração informações típicas de sistemas ubíquos (e.g., informações contextuais).

É importante ressaltar que, dependendo do sistema a ser desenvolvido, os atributos mencionados anteriormente podem ter pesos diferentes em relação a confiança no funcionamento. Além disso, é possível que, em certos sistemas, os atributos sejam vistos como antagônicos (e.g., disponibilidade e proteção). Logo, é importante fazer uso de um conjunto de métodos ou técnicas para garantir que o sistema resultante seja de confiança. De acordo com a literatura, os meios para alcançar a confiança no funcionamento são:

- Prevenção de Falhas: evitar a ocorrência ou introdução de falhas.
- Tolerância a Falhas: fornecer um serviço de acordo com a especificação, apesar da ocorrência de falhas.
- Supressão de Falhas: como reduzir a presença (e.g. número, gravidade) de falhas.

- Previsão de Falhas: como estimar o corrente número, a incidência futura e as consequências das falhas.

De acordo com (ALVES E VASQUEZ 1998), a prevenção e tolerância podem ser vistas como meios para a obtenção da confiança no funcionamento (e.g. dotar o sistema com a capacidade de prestar um serviço de acordo com a especificação) e a remoção e previsão de falhas podem ser vistas como meios para a validação da confiança no funcionamento (e.g. alcançar confiança na capacidade de prestar um serviço de acordo com a especificação). O conjunto de conceitos mencionados até então, referentes à confiança no funcionamento, é ilustrado na Figura 8.



**Figura 10: Árvore da Confiança no Funcionamento adaptada de (AVIZIENIS et al 2004)**

Como pode-se observar na figura, os meios para garantir a confiança no funcionamento são: prevenção de falhas, previsão de falhas, remoção de falhas e tolerância a falhas. Segundo (ALVES e VASQUEZ 1998), esses meios são objetivos que não podem ser alcançados em sua totalidade pois, as atividades relacionadas a eles são executadas por seres humanos logo, são susceptíveis a erros.

Diante desse contexto, os autores sugerem que é necessária uma utilização combinada desses meios para que seja possível conceber e implementar um sistema computacional no qual se possa justificadamente confiar.

A motivação para esta afirmação está baseada nas afirmações de (ALVES e VASQUEZ 1998) onde, segundo os autores, apesar da prevenção de falhas ser parcialmente alcançada por intermédio de metodologias de concepção e regras (e.g. técnicas para elicitación de requisitos), há a necessidade da remoção de falhas visto que estas ainda podem surgir.

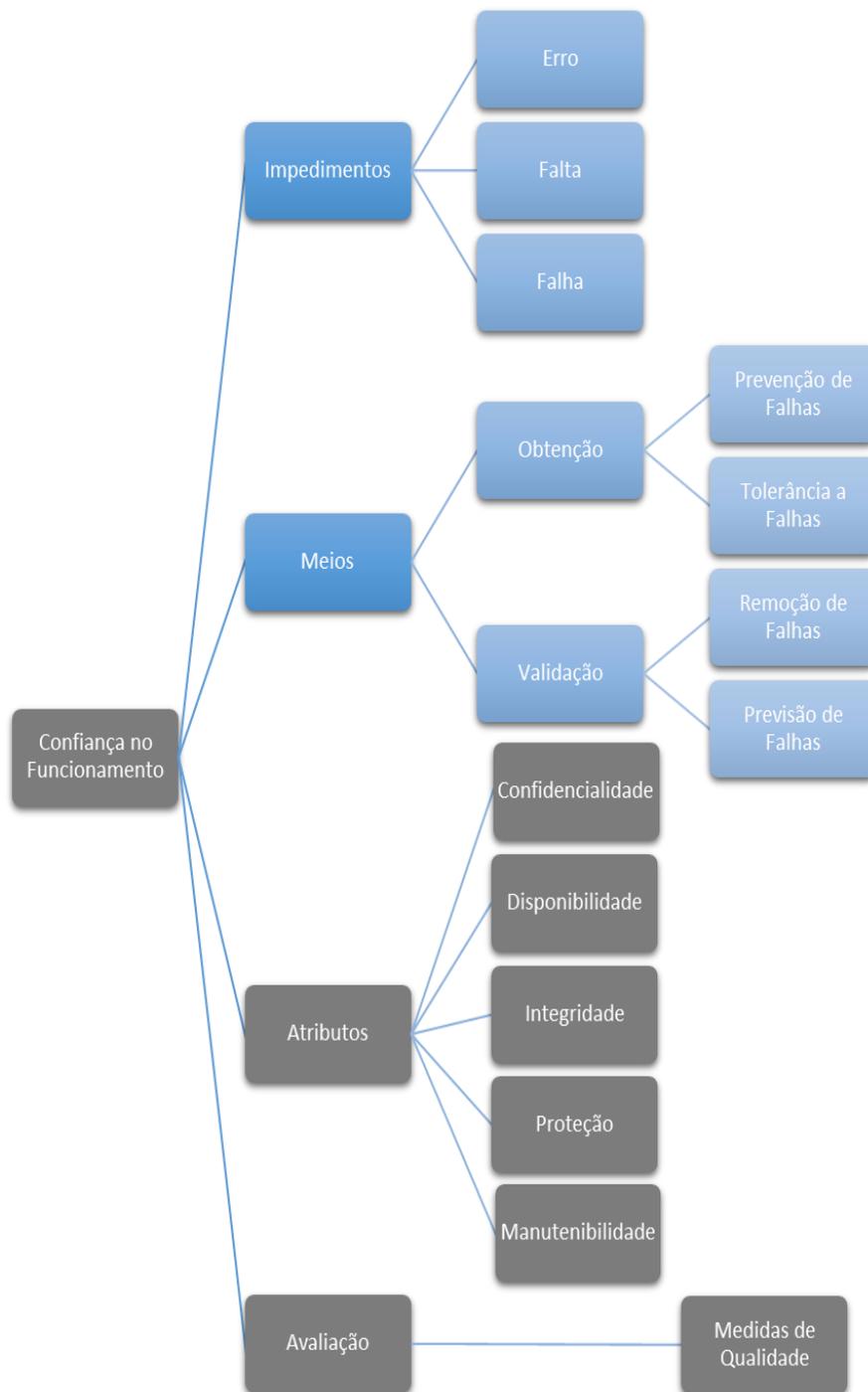
Porém, a remoção de todas as falhas de um sistema é impossível, visto que a ocorrência destas podem aparecer após a fase de projeto e implementação. Ocasionalmente, conseqüentemente, na necessidade da previsão de falhas. Além disso, a crescente dependência associada aos sistemas computacionais implica, muitas vezes, no uso de mecanismos de tolerância a falhas.

Logo, como alternativa para a garantia da confiança no funcionamento, visto que os meios para atingi-la são dependentes e de combinações complexas e flexíveis, dependendo do sistema ser desenvolvido, é possível fazer uso dos atributos mencionados por (AVIZIENIS et al 2004) como critérios para avaliações da qualidade. Com essas avaliações é possível então guiar o processo de desenvolvimento do sistema de forma a concebe-lo como algo justificadamente confiável.

A árvore da confiança no funcionamento, resultante dos desafios e informações levantadas com a revisão bibliográfica é apresentada na Figura 11. Onde é dado destaque a avaliação dos atributos da confiança no funcionamento por meio de medidas de qualidade de software.

## **2.4 Considerações Finais**

Nesse capítulo foi apresentada uma visão geral sobre as áreas de pesquisa relacionadas a este trabalho: Sistemas Ubíquos, Confiança no Funcionamento de Sistemas de Software e Qualidade de Software.



**Figura 11: Árvore da Confiança Resultante da Fundamentação Teórica**

Com o levantamento literário feito, é possível afirmar que, para desenvolvermos um sistema de software no qual seja possível justificadamente confiar é necessário levar em consideração uma série de atributos que permitem avaliar a qualidade do sistema.

Essa avaliação pode ser feita por meio do uso de medidas de qualidade de software. Porém, as medidas de qualidade, para serem aplicadas para o domínio de sistemas ubíquos

para dispositivos móveis, devem levar em consideração, no momento de sua definição, características particulares desse tipo de sistema (e.g. sensibilidade ao contexto).

Diante desse cenário, é possível identificar a carência de medidas aplicáveis para esse domínio específico, a qual pode ser suprida por meio da definição de novas medidas, a qual pode ser feita com o uso da metodologia GQM.

Os trabalhos encontrados na literatura, relacionados a este, são apresentados em detalhes no Capítulo 3.

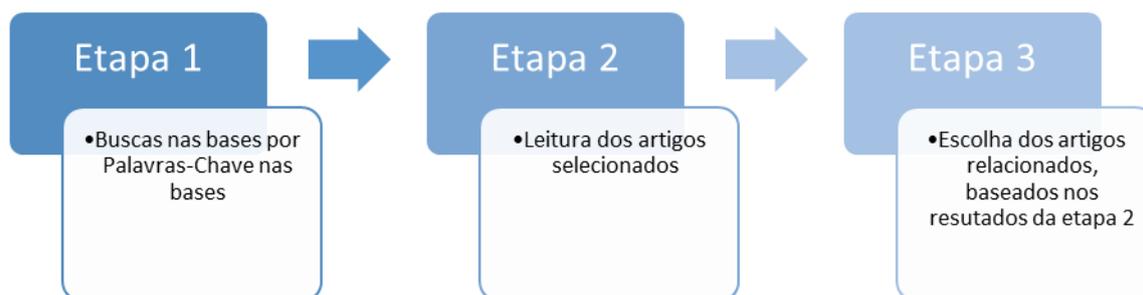
# Capítulo 3

## Trabalhos Correlatos

Neste capítulo é apresentada uma visão comparativa dos trabalhos relacionados a esta pesquisa. A Seção 3.1 apresenta a metodologia de busca utilizada e a Seção 3.2 apresenta uma análise dos trabalhos encontrados, comparando-os com o trabalho a ser proposto nesta dissertação e, por fim a Seção 3.3 apresenta as considerações finais, com base na comparação feita.

### 3.1 Metodologia de Busca

O processo de identificação dos trabalhos relacionados, realizado nesta pesquisa, ocorre em três etapas, considerando as bases eletrônicas da IEEE, ACM e Springer. O processo de filtragem dos artigos é representado na Figura 12 e detalhado no restante deste capítulo.



**Figura 12: Processo de identificação dos trabalhos relacionados**

Como observado na Figura 12, a etapa 1 é caracterizada pela busca por palavras-chave nas bases eletrônicas citadas anteriormente. As palavras-chave são derivadas do

*IEEE - <http://ieeexplore.ieee.org/Xplore/home.jsp>*

*ACM - <http://dl.acm.org/>*

*Springer - <http://www.springer.com.br/>*

objetivo da proposta. Primeiro são identificados os objetos de interesse (características e medidas de qualidade), depois o propósito (avaliação da confiança no funcionamento) e,

por fim, o contexto (sistemas ubíquos ou sistemas ubíquos para dispositivos móveis). Durante a busca, foram considerados também sinônimos, traduções para o inglês e palavras alternativas, apresentados na Tabela 4:

**Tabela 4: Palavras-chave utilizadas para a busca de trabalhos**

Objeto	Propósito	Contexto
<i>Características</i>		
<i>Subcaracterísticas</i>		<i>Sistemas Sensíveis ao Contexto</i>
<i>Medidas</i>	<i>Avaliar a confiança no funcionamento</i>	<i>Sistemas Pervasivos</i>
<i>Métricas</i>		<i>Sistemas Ubíquos</i>
<i>Crítérios</i>		<i>Sistemas móveis</i>
<i>Atributos</i>		
<i>Fatores</i>		
<i>Modelo</i>		
<i>Indicadores</i>		

O intervalo de tempo considerado é de 1991 até 2015, tendo como base para a escolha o surgimento do conceito de computação ubíqua (1991) até os trabalhos atuais. Foram então encontrados 121 artigos considerados relevantes, publicados em conferências e periódicos da área.

Ainda de acordo com a Figura 7, na segunda etapa, é realizada a leitura completa de cada um dos artigos, de onde é possível extrair: referência, problema abordado, contribuição ou inovação do artigo, semelhanças e divergências entre o artigo e o trabalho em desenvolvimento e, por fim, desafios encontrados.

Na terceira e última etapa, com base nas informações coletadas na etapa 2, foram selecionados 14 trabalhos considerados pertinentes a este, descritos de maneira resumida na Tabela 5:

**Tabela 5: Referência e descrição dos trabalhos relacionados**

<i>Referência</i>	<i>Descrição Resumida</i>
<i>Scholtz and Consolvo, 2004</i>	<i>Propõem um framework para avaliação de aplicações ubíquas</i>
<i>AVIZIENIS et al. 2004</i>	<i>Propõem uma taxonomia para confiança no funcionamento (dependability).</i>
<i>Ranganathan, 2005</i>	<i>Propõe um benchmark para avaliação de medidas e atividades em ambientes da computação ubíqua.</i>
<i>Stajano, 2008</i>	<i>Aborda questões e desafios relacionados à segurança em sistemas ubíquos.</i>
<i>Jia, 2009</i>	<i>Propõe um framework para avaliação da confiança baseado em controle de</i>

	<i>acesso nos processos de interação humano computador.</i>
<i>Abi-Char, 2010</i>	<i>Propõe um framework para garantia de segurança e privacidade em sistemas sensíveis ao contexto.</i>
<i>Karvonen 2010</i>	<i>Analisa diferentes aspectos da confiança humana que afetam o uso de sistemas ubíquos inteligentes de transporte. O objetivo é contribuir para as bases conceituais da confiança na interação humano-computador.</i>
<i>Treck 2011</i>	<i>Avalia uma metodologia qualitativa de avaliação dinâmica para o gerenciamento de confiança em sistemas ubíquos, baseado em simulações</i>
<i>iso 25010</i>	<i>Propõe características, subcaracterísticas e medidas para a avaliação da qualidade em sistemas de software de maneira geral.</i>
<i>Lee and Yun, 2012</i>	<i>Propõem medidas para a avaliação da usabilidade em serviços ubíquos, tendo como base as interações e serviços.</i>
<i>Santos, 2013</i>	<i>Propõe um modelo de qualidade para avaliação de sistemas ubíquos.</i>
<i>Hofmman and Sollner, 2014</i>	<i>Propõem uma abordagem para aumentar a confiança durante o processo de desenvolvimento de sistemas ubíquos utilizando conceitos e idéias sobre teoria da confiança comportamental.</i>
<i>Sherchan et al. 2014</i>	<i>Apresentam a primeira revisão abrangente da literatura social e ciência da computação na confiança (trust) em redes sociais.</i>
<i>Liu et al. 2014</i>	<i>Apresentam um modelo que descreve o papel da personalização e controle na avaliação de custos e benefícios associados à divulgação de informações privadas dos usuários em sistemas pervasivos.</i>

Apesar de antigas, algumas das referências como (AVIZIENIS et al 2004) e (RANGANATHAN 2005), são de extrema importância pois, além de fundamentarem os trabalhos seguintes trazem contribuições essenciais para o trabalho em desenvolvimento nesta proposta de dissertação que vão além da fundamentação teórica, como, por exemplo, a taxonomia utilizada para a confiança no funcionamento apresentada em (AVIZIENIS et al 2004) ou as medidas apresentadas por (SCHOLTZ e CONSOLVO 2004) e (RANGANATHAN 2005).

## **3.2 Análise dos Trabalhos Correlatos**

A análise dos trabalhos correlatos leva em consideração os seguintes aspectos: as vantagens encontradas em cada um dos trabalhos, as desvantagens e as relações entre eles e esta

pesquisa de mestrado. O objetivo da análise é, com base nos aspectos considerados, identificar quais são as características que influenciam na confiança no funcionamento de sistemas ubíquos executando em dispositivos móveis e qual a metodologia utilizada para a sua avaliação. No caso da identificação de medidas de qualidade para a avaliação, é também analisada a adequação dessas medidas para o domínio em questão.

Buscando alcançar o propósito definido, três questões, definidas por (SANTOS 2014) e adaptadas para esta pesquisa, são consideradas durante a análise dos trabalhos relacionados:

1. Quais metodologias, técnicas, procedimentos (e.g. modelo de qualidade, medidas de qualidade, framework ou outras) são utilizadas para avaliar a confiança no funcionamento em sistemas ubíquos?

Para responder esta questão é necessário informar a existência de diferentes métodos, procedimento e metodologias existentes na área de qualidade de software que podem ser usados para a avaliação de atributos de qualidade. Dentre esses, é possível citar: modelos de qualidade, medidas de qualidade, framework de medidas, framework de características, processos, análise de requisitos e outros (SANTOS 2014).

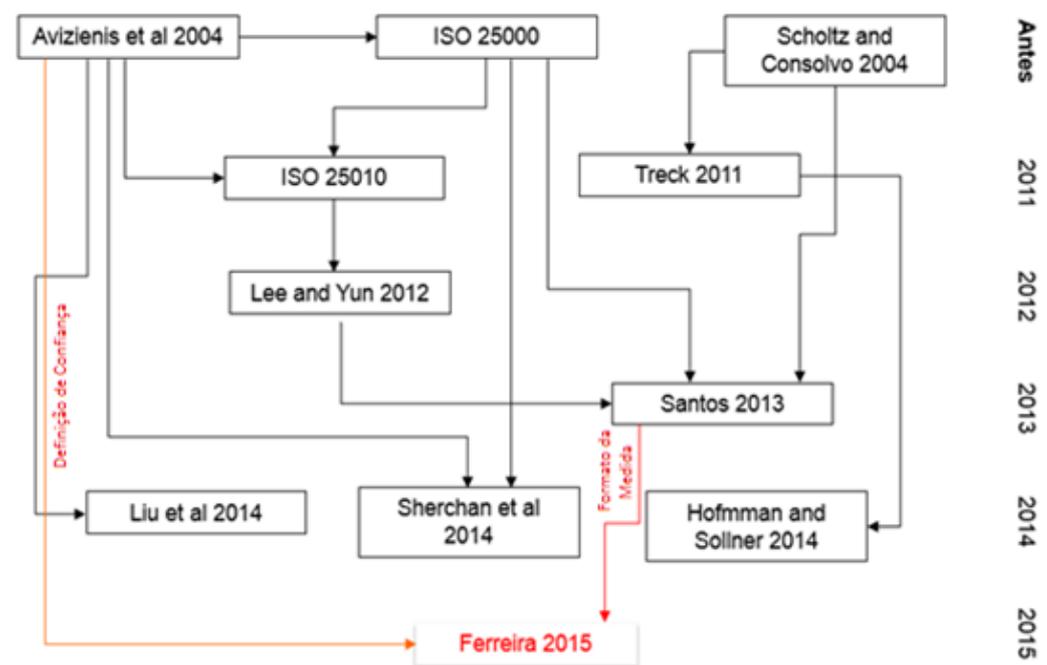
Durante a análise dos trabalhos é possível identificar três metodologias distintas para a avaliação da confiança no funcionamento: framework de características, medidas e características, detalhadas a seguir:

- Framework de características – nesse caso, o trabalho apresenta as características e/ou subcaracterísticas de qualidade como uma lista de fatores que devem ser avaliados em um sistema ubíquo (SCHOLTZ e CONSOLVO 2004) (LEE e YUN 2012) (JIA 2009) (ABI CHAR 2010).
- Medidas de Qualidade de *Software* – nesse caso, o trabalho apresenta uma lista de medidas organizadas dentro de características de qualidade (RANGANATHAN 2005) (ISO 25010) (SANTOS 2013) (SANTOS 2014).
- Características de Avaliação – nesse tipo de trabalho é apresentado um conjunto de características que podem ser consideradas em uma avaliação envolvendo sistemas ubíquos, mas não está explicitamente definido como a avaliação pode ser realizada (STAJENO 2008) (KARVONEN 2010) (TRECK

2011) (HOFMAN e SOLLNER 2014) (SCHERCHAN et al 2014) (LIU et al 2014).

- Quais características (atributos ou fatores) de qualidade são utilizadas para avaliar a confiança no funcionamento em sistemas ubíquos?

Buscando responder à Questão 2, a análise das características de qualidade encontradas nesses trabalhos é apresentada na Figura 13, a seguir:



Onde A → B indica que a definição de confiança utilizada por B foi proposta por A

Figura 13: Relacionamento entre os trabalhos relacionados

O fluxograma apresentado na Figura 13 relaciona os trabalhos de acordo com os conceitos utilizados por cada um deles (e.g., confiança (do inglês *trust*) ou confiança no funcionamento (do inglês *dependability*)). Em outras palavras, as relações expressas no fluxograma levam em consideração as características de qualidade utilizadas por cada trabalho para a avaliação da confiança em sistemas de software.

É possível perceber que os conceitos que fundamentam os demais trabalhos são encontrados em (AVIZIENIS et al 2004), (SCHOLTZ e CONSOLVO 2004) e (ISO 25000).

A partir do conceito definido nos três trabalhos iniciais, outros trabalhos foram fundamentados (e.g., (LEE e YUN 2012), (LIU et al 2014)). Para o trabalho fruto desta

dissertação, a definição adotada para confiança no funcionamento pode ser encontrada em (AVIZIENIS et al 2004).

A escolha da definição baseia-se na importância diante dos trabalhos relacionados, além disso, os atributos considerados por (AVIZIENIS et al 2004) abrange grande parte das possíveis características que impactam o usuário em relação a sua confiança no funcionamento do sistema. Segundo (SCHOLTZ e CONSOLVO 2004), apenas a privacidade, consciência e controle são considerados como impactantes para a avaliação da confiança, não incluindo a disponibilidade do sistema e a integridade dos dados. Esta ausência de características reforça a melhor adequação da definição de (AVIZIENIS et al 2004) para os objetivos desta pesquisa.

Além disso, o formato utilizado para a definição das medidas, bem como a metodologia utilizada para a definição é baseada na (ISO 25000). O estabelecimento de um formato padrão para a definição das medidas é de fundamental importância, pois possibilita precisão e clareza para a realização das medições. O formato utilizado pela (ISO 25000) é considerado o mais apropriado para ser adotado nesta pesquisa, pois a presença de uma função de medição e valores de interpretação são partes obrigatórias da medida. Com isso, as medidas definidas seguindo esse formato podem ser utilizadas como uma metodologia de avaliação de qualidade repetível e aplicável quando utilizadas em conjunto com outras informações apresentadas no guia presente no Apêndice A.

3. Existem medidas de qualidade de *software* para avaliar a confiança no funcionamento nesses sistemas? É necessário adaptá-las?

Em resposta a terceira questão, é possível encontrar medidas de qualidade de software para a avaliação da confiança nos trabalhos de (SCHOLTZ e CONSOLVO 2004), (RANGANATHAN 2005), (SANTOS 2013), (LEE e YUN 2012) e (ISO 25010). Os principais problemas relacionados às medidas encontradas na literatura são a falta de um formato padrão para a definição destas e o foco específico de cada um dos trabalhos quanto aos atributos (ou características) considerados para a avaliação da confiança no funcionamento.

Em geral, é possível perceber a restrição da avaliação apenas a um conjunto específico dos atributos de impacto, apontados no próprio trabalho, desconsiderando os demais. Isto implica na necessidade da definição de novas medidas.

De acordo com (SCHOLTZ e CONSOLVO 2004), a confiança pode ser avaliada considerando questões relativas à privacidade, controle das informações pessoais do usuário e consciência do usuário sobre o uso de suas informações por parte do sistema. As medidas definidas em seu trabalho são associadas somente a proteção (*safety*), não considerando, portanto, atributos como confiabilidade, disponibilidade e confiabilidade.

O mesmo pode ser aplicado para o trabalho encontrado em (RANGANATHAN 2005), porém, nesse caso, o foco das medidas definidas está na confiabilidade e proteção, desconsiderando os demais atributos. Além disso, as medidas definidas pelo autor não seguem uma especificação precisa (e.g., com nome, função de medição e método de coleta) e são apresentadas apenas com nome e descrição, não sendo possível coletá-las por falta de funções de medição.

Dessa forma, utilizando as medidas propostas por (RANGANATHAN 2005) (e.g., a taxa de erros de contexto inferidos), pode não ser possível obter uma medição repetível, pois a falta de função de medição pode possibilitar resultados com um nível de abstração e ambiguidade significativos. Além disso, segundo o autor, o conjunto de medidas definidas em seu trabalho auxilia a avaliação da confiança no funcionamento, porém não é o suficiente para o domínio dos sistemas ubíquos, acarretando na necessidade da definição de outras medidas.

O trabalho de (SANTOS 2014) possui as mesmas características do trabalho de (RANGANATHAN 2005) em relação a definição das medidas, as quais são apresentadas para avaliação da confiança, mas possuem apenas nome e descrição, pois, a definição destas, na pesquisa do autor, é apresentada como um trabalho futuro.

Assim, as medidas definidas nesta pesquisa de mestrado podem ser vistas como uma contribuição para a evolução do trabalho realizado por (SANTOS 2014). Além disso, as medidas definidas neste trabalho auxiliam também para a evolução da (ISO 25010) e de (RANGANATHAN 2005) pois, levam em consideração aspectos dos sistemas ubíquos não encontrados nos

Na (ISO 25010) é possível encontrar três categorias distintas de medidas: internas, que analisam o comportamento estático, externas, que analisam o comportamento do sistema sob condições específicas e, em uso. As medidas internas e externas estão associadas a avaliação dos requisitos e detalhes da qualidade do produto que podem ser melhoradas durante a implementação, revisão e teste, já as medidas para avaliar a qualidade em uso estão associadas a visão do usuário da qualidade do produto de software quando utilizado em ambiente e contexto de uso estabelecidos.

A análise comparativa entre os trabalhos relacionados, baseada no que foi exposto ao longo deste capítulo, pode ser encontrada de forma sintetizada na seção a seguir.

### **3.3 Considerações Finais**

Com base no que foi apresentado nas seções anteriores deste capítulo e com a análise comparativa apresentada na Tabela 6, é possível observar que a principal contribuição dos trabalhos relacionados é a definição da taxonomia utilizada para a confiança no funcionamento e o auxílio para a definição de medidas, pois é possível extrair de alguns trabalhos medidas que podem ser adaptadas. As desvantagens concentram-se na maior parte em medidas definidas sem um padrão específico. Além disso, é possível perceber que alguns dos trabalhos desconsideram características particulares dos sistemas ubíquos (AVIZIENIS et al 2004) (JIA 2009).

A principal contribuição deste trabalho está, portanto, em definir medidas de qualidade de *software* para a avaliação da confiança no funcionamento que reflitam as características dos sistemas ubíquos e considerem os atributos da definição de confiança adotada. Além deste, outros desafios de pesquisa e questões em aberto são identificados durante a revisão da literatura, levando em consideração os aspectos apresentados na Seção 3.1 e são apresentados na Tabela 7.

**Tabela 6: Análise comparativa dos trabalhos relacionados**

Referência	Considera o domínio dos sistemas ubíquos	Avalia por meio de medidas de qualidade	As medidas são definidas seguindo formato padrão	Existe algum guia que auxilie a aplicação das medidas	Considera a disponibilidade como fator de impacto	Considera a confiabilidade como fator de impacto	Considera a manutenibilidade como fator de impacto	Considera a integridade como fator de impacto	Considera a proteção como fator de impacto
Scholtz and Consolvo, 2004	SIM	SIM	NÃO	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
Avizienis et al. 2004	NÃO	NÃO	NÃO	NÃO	SIM	SIM	SIM	SIM	SIM
Ranganathan, 2005	SIM	SIM	NÃO	NÃO	NÃO	SIM	NÃO	SIM	SIM
Stajano, 2008	SIM	NÃO	NÃO	NÃO	SIM	SIM	NÃO	SIM	SIM
Jia, 2009	SIM	NÃO	NÃO	NÃO	NÃO	SIM	NÃO	SIM	SIM
Abi-Char, 2010	SIM	NÃO	NÃO	NÃO	NÃO	SIM	NÃO	SIM	SIM
Karvonen 2010	SIM	NÃO	NÃO	NÃO	SIM	SIM	SIM	SIM	SIM
Treck 2011	SIM	NÃO	NÃO	NÃO	SIM	NÃO	NÃO	SIM	SIM
ISO 25010	PARCIAL	SIM	SIM	SIM	NÃO	NÃO	NÃO	NÃO	SIM
Lee and Yun, 2012	SIM	SIM	SIM	NÃO	NÃO	SIM	NÃO	NÃO	SIM
Santos, 2013	SIM	PARCIAL	PARCIAL	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
Hofmman and Sollner, 2014	SIM	NÃO	NÃO	NÃO	NÃO	SIM	NÃO	SIM	SIM
Sherchan et al. 2014	SIM	NÃO	NÃO	NÃO	NÃO	SIM	NÃO	NÃO	SIM
Liu et al. 2014	SIM	SIM	NÃO	NÃO	NÃO	SIM	NÃO	NÃO	SIM

Onde PARCIAL indica que o trabalho leva em consideração o aspecto em questão, porém não de maneira tão ampla como este trabalho

**Tabela 7: Questões e desafios encontrados**

Referência	Questões em aberto / Desafios encontrados
Scholtz e Consolvo, 2004	<ul style="list-style-type: none"><li>• Como garantir que o processamento das informações pessoais seja de fundamental importância para a confiança no sistema?</li><li>• Como personalizar a privacidade de acordo com o usuário e/ou dispositivo?</li><li>• Como garantir que um usuário intruso não interfira nas informações de outro?</li><li>• Como equilibrar o <i>trade-off</i> entre conhecimento de informações e privacidade?</li></ul>
Nixon, 2005	<ul style="list-style-type: none"><li>• Como garantir controle de acesso, gerenciar identidades e lidar com os aspectos sociais e legais em ambientes inteligentes?</li><li>• Como prover interações seguras em ambientes inteligentes?</li><li>• Como estabelecer abertura e preservar a privacidade em ambientes inteligentes?</li></ul>
Ranganathan, 2005	<ul style="list-style-type: none"><li>• Quais as características em comum aos ambientes ubíquos e outros ambientes que devem ser levadas em consideração para medição da segurança e privacidade nos ambientes inteligentes?</li><li>• Como expressar a probabilidade do contexto ter sido deduzido de maneira correta?</li></ul>
Stajano, 2008	<ul style="list-style-type: none"><li>• Como expressar erros de acurácia com relação ao contexto?</li><li>• Como identificar os pontos fracos da segurança do sistema?</li><li>• Quais as partes mais valiosas do sistema?</li><li>• Onde um possível ataque causaria dano máximo?</li><li>• Quais as motivações para atacar o sistema?</li><li>• Como sobrecarregar o sistema ao ponto de gerar uma falha?</li><li>• É possível efetuar um ataque em grande escala, atingindo vários nós do sistema?</li></ul>
Jia, 2009	<ul style="list-style-type: none"><li>• Como garantir que uma entidade maliciosa não tenha acesso a informações que não são permitidas?</li><li>• Como garantir que entidades maliciosas não manipulem informações?</li><li>• Como garantir que uma entidade maliciosa não aja fornecendo informações erradas para os usuários?</li><li>• Como garantir que comportamentos conflitantes entre diferentes entidades do sistema não gerem uma falha de segurança?</li></ul>

# Capítulo 4

## Medidas de Qualidade de Software

Este capítulo apresenta medidas de qualidade de *software* para a avaliação da confiança no funcionamento em cenários de sistemas ubíquos que executam em dispositivos móveis. Para a definição dessas medidas, a metodologia *Goal-Question-Metrics* (GQM) é utilizada. As Seções 4.1, 4.2, 4.3 e 4.4 apresentam, respectivamente, as fases de planejamento, definição, coleta e interpretação, sendo estas correspondentes às etapas do GQM. Por fim, a Seção 4.5 apresenta as considerações finais sobre as medidas definidas.

### 4.1 Planejamento

Como mencionado no Capítulo 2, a medição de *software* é uma avaliação quantitativa que permite o melhor entendimento sobre os processos e produtos de software. Segundo (BASILLI et al. 2002) (ROCHA et al. 2012), realizando medições é possível auxiliar o planejamento, controle e melhoria de processos e produtos. Ainda segundo os autores, o elemento básico da medição é chamado de medida.

Dando continuidade ao trabalho de (SANTOS 2014), esta dissertação também utiliza a metodologia GQM para a definição de medidas de qualidade de software. Estas medidas são utilizadas para avaliar as características de qualidade dos sistemas ubíquos nos cenários mencionados anteriormente, em específico a característica de confiança no funcionamento desse tipo de sistema.

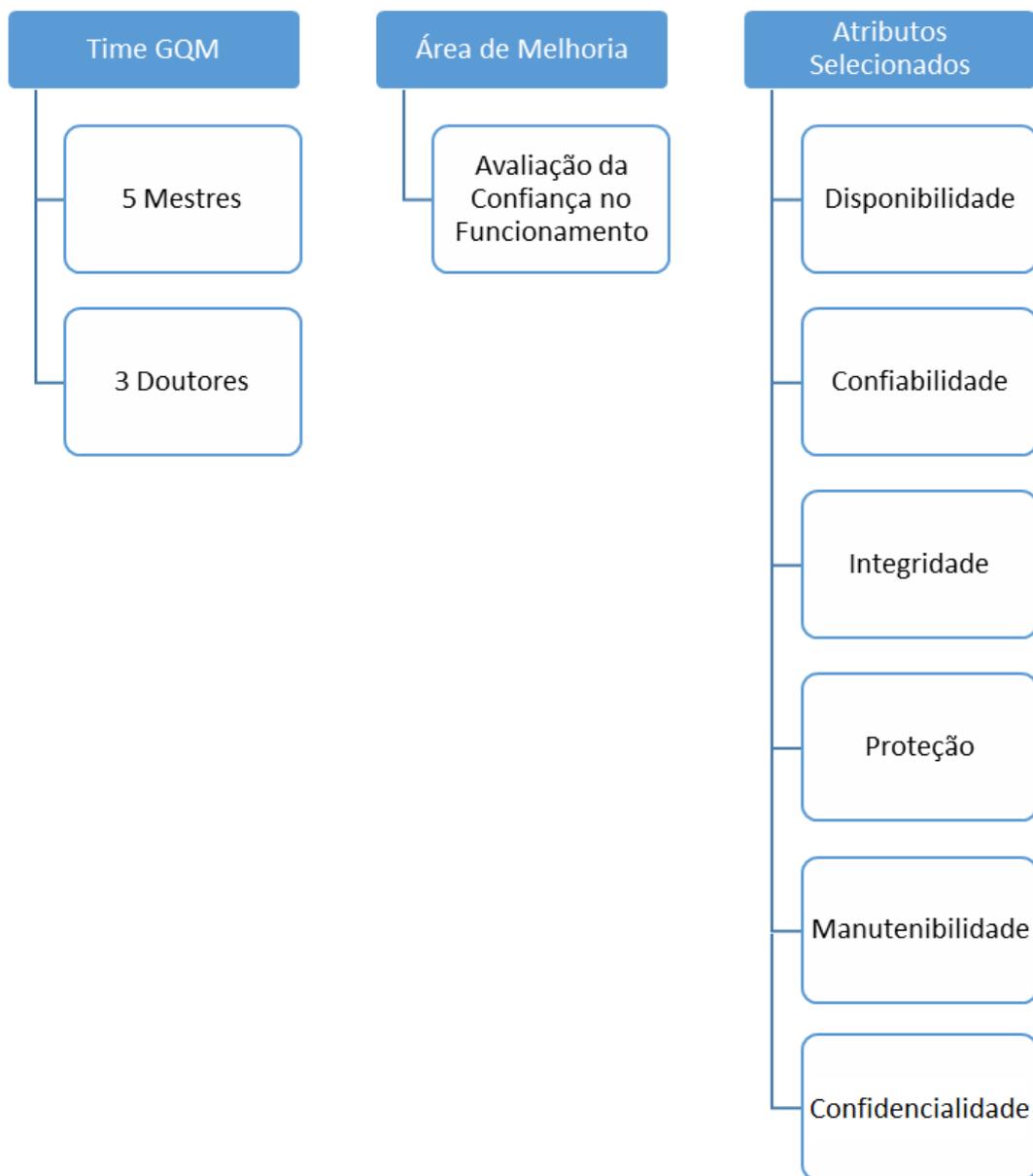
Assim como no trabalho de (SANTOS 2014), o objetivo principal da fase de planejamento deste trabalho é estabelecer um time GQM, uma área de melhoria, um projeto de aplicação e, o time do projeto (e.g., todas as pessoas envolvidas e interessadas no processo de definição).

Para esta pesquisa, o time GQM foi definido de acordo com as áreas de interesse apresentadas no Capítulo 2. Foram selecionados especialistas com conhecimento nas áreas de sistemas ubíquos, qualidade de *software* e confiança em sistemas de *software*.

Um participante é considerado especialista se possui mestrado (completo) ou doutorado (completo ou em andamento) em pelo menos uma das áreas de interesse. No total, oito especialistas participaram das entrevistas GQM, sendo estas realizadas de maneira individual.

Além disso, com base na motivação e objetivos desta pesquisa, foi possível estabelecer uma área de melhoria: a avaliação da confiança no funcionamento em cenários de sistemas ubíquos que executam em dispositivos móveis. O projeto de aplicação e o time de projeto não foram definidos, pois, assim como em (SANTOS 2014), não foi considerado um projeto ou sistema específico, mas sim atributos de qualidade dos sistemas ubíquos (e.g., disponibilidade, confiabilidade, integridade, proteção, confidencialidade e manutenibilidade).

A etapa de planejamento é ilustrada na Figura 14.



**Figura 14: Planejamento do GQM**

## 4.2 Definição

Como apresentado no Capítulo 2, a fase de definição de medidas é composta por:

- Definição do objetivo de medição, com objeto, propósito e foco;
- Entrevistas GQM com especialistas;
- Utilização dos resultados das entrevistas para derivar questões; e, por fim,
- Medidas.

Para a definição do objetivo de medição é utilizado o *template* definido por (BASILLI et al. 2002), apresentado no Capítulo 2.

No caso dessa pesquisa, o objeto sob medição é o sistema ubíquo que executa em dispositivos móveis, o propósito é avaliar e os focos de qualidade são os seis atributos de qualidade selecionados anteriormente: Disponibilidade, Integridade, Confiabilidade, Manutenibilidade, Confidencialidade e Proteção. Por fim, o ponto de vista é o do usuário e o do desenvolvedor.

Assim como indica (BASILI et al 2002), após a definição do objetivo de medição, é necessário realizar as entrevistas GQM com o time estabelecido na fase de planejamento. Essas entrevistas são fundamentais para identificar quais são as questões relevantes, com respeito ao objetivo e, quais as possíveis medidas capazes de respondê-las.

Como mencionado anteriormente, as entrevistas GQM foram realizadas individualmente, totalizando oito reuniões com oito especialistas distintos:

- 2 mestrandos e 3 doutorandos na área de sistemas ubíquos
- 2 mestres e 1 doutorando na área de segurança e confiança.

A divisão ocorreu, pois, as reuniões dependiam da disponibilidade dos entrevistados, não sendo possível entrevistar dois ou mais especialistas de maneira simultânea.

Para cada pessoa ou grupo, as atividades realizadas foram:

- apresentação do problema, motivação e objetivos da pesquisa;
- análise da lista de medidas encontradas na literatura (ou definidas com base nas medidas encontradas); e,
- a seleção de medidas e sugestões, por parte dos especialistas, de novas medidas não apresentadas durante a reunião.

Portanto, essas atividades estão em conformidade com o que sugere (BASILI et al 2002) e com o estudo realizado por (SANTOS 2014).

Para cada foco de qualidade apresentado foi utilizado um *abstract sheet* (Tabela 8) com o intuito de facilitar a entrevista e documentação das medidas.

**Tabela 8: Abstract Sheet**

<p><b>Foco de qualidade</b></p> <p><i>Qual é o conhecimento atual dos especialistas com respeito ao foco de qualidade?</i></p>	<p><b>Fatores de variação</b></p> <p><i>Quais fatores dos sistemas ubíquos os especialistas acham que influenciam esse fator?</i></p>
<p><b>Hipóteses de <i>baseline</i></b></p> <p><i>Quais são as possíveis medidas para mensurar o objeto de estudo de acordo com os especialistas?</i></p> <p><i>Qual é o conhecimento atual dos especialistas com respeito a essas medidas?</i></p>	<p><b>Impacto nas hipóteses de <i>baseline</i></b></p> <p><i>Quais tipos de dependências existem entre as medidas e os fatores de variação?</i></p>

Com base nas informações obtidas com a revisão da literatura e com as entrevistas GQM foi possível derivar as seguintes questões:

- Q1 - Qual o grau de confiança do sistema?
  - Q1.1- Qual o grau de disponibilidade do sistema?
  - Q1.2 - Qual o grau de integridade do sistema?
  - Q1.3 - Qual o grau de confiabilidade do sistema?
  - Q1.4 - Qual o grau de manutenibilidade do sistema?
  - Q1.5 - Qual o grau de proteção do sistema?
  - Q1.6- Qual o grau de confidencialidade do sistema?

Seguindo a definição dos atributos de confiança no funcionamento, definidos por (AVIZIENIS et al 2004) e mencionados nas questões Q1.1, Q1.2, Q1.3, Q1.4, Q1.5 e Q1.6. A questão Q1 busca identificar o grau justificado de confiança no funcionamento do sistema. A confiança é justificada com base na disponibilidade, integridade, confiabilidade, manutenibilidade e proteção do sistema, implicando, portanto, nas questões.

A questão Q1.1 busca identificar se o sistema está disponível sempre que requisitado. A questão Q1.2, por sua vez, busca identificar a ausência de alterações inadequadas no sistema. A questão Q1.3 busca avaliar se o sistema funciona conforme o especificado. A questão Q1.4 tem

por objetivo avaliar o grau de eficácia e eficiência com o qual o sistema pode ser modificado. A questão Q1.5 avalia o grau em que o sistema pode levar a um estado no qual a vida humana, a saúde, a propriedade ou o ambiente está em perigo. Por fim, a questão Q1.6 busca avaliar a proteção da informação ou serviço contra acessos não autorizados.

As medidas resultantes seguem, a princípio, o formato da norma ISO 25000 e são compostas, em sua essência, por identificador, nome, definição, função de medição e método de coleta. Apresentada no Capítulo 2, essa norma é referência quando se trata de medidas de qualidade de software.

O conjunto resultante das medidas definidas neste trabalho é descrito na Tabela 9. Nesta tabela são apresentados:

- Identificador: sigla única capaz de identificar a medida;
- Nome: nome da medida;
- Definição: descrição da medida;
- Função de Medição: fórmula utilizada para a medição;
- Interpretação: intervalo de valores esperados ou aceitáveis;
- Método de Coleta: indicação do método ou procedimento a ser utilizado para a coleta da medida;
- Referência: a referência da medida pode ser algum trabalho encontrado na literatura (pois a medida já se encontrava formatada de acordo com a [ISO/IEC 25000]), ou uma adaptação (medida encontrada na literatura e que possui apenas nome e descrição, ou nome e função de medição, necessitando, portanto, de uma adaptação). Ou, por fim, fruto de uma entrevista GQM (sugestão de um especialista ou definição com base em outras medidas);
- Subcaracterística associada: atributo(s) da confiança no funcionamento que está(ão) diretamente associado(s) a medida.

Como mencionado no Capítulo 2, as medidas de qualidade de software devem ser definidas de forma a tornar a atividade de avaliação algo repetível e aplicável. Para tanto, segundo (ROCHA et al 2012), são necessárias diversas informações (e.g. momento da coleta, periodicidade da coleta, responsável pela coleta). As demais informações, exemplificadas acima, podem ser encontradas no Guia apresentado no Anexo A.

Neste guia, as medidas definidas neste trabalho, seguem um formato adaptado de (ROCHA et al 2012). Contendo:

- Nome: nome da medida;

- Definição: descrição sucinta da medida;
- Mnemônico: sigla para identificar a medida, identificador;
- Tipo de Medida: classificação da medida quanto a sua dependência funcional, podendo ser uma medida base ou derivada;
- Entidade Medida: entidade que a medida mede;
- Propriedade Medida: propriedade da entidade medida que é quantificada pela medida;
- Unidade da Medida: unidade de medida em relação à qual a medida é medida;
- Tipo de Escala: natureza dos valores que podem ser atribuídos à medida;
- Valores da Escala: valores que podem ser atribuídos à medida;
- Intervalo Esperado dos Dados: limite de valores da escala definida;
- Fórmula de Medição: fórmula utilizada no procedimento de medição;
- Procedimento de Coleta: descrição do procedimento que deve ser realizado para coletar uma medida;
- Momento da Coleta: momento no qual a medida deve ser coletada;
- Periodicidade da Coleta: frequência de coleta da medida;
- Responsável pela Coleta: papel desempenhado pelo recurso humano responsável pela coleta da medida;
- Procedimento de Análise: descrição do procedimento que deve ser realizado para analisar uma medida;
- Momento da Análise: momento no qual a medida deve ser analisada;
- Periodicidade da Análise: frequência de análise da coleta da medida;
- Responsável pela Análise: papel desempenhado pelo recurso humano responsável pela análise dos resultados.

É importante ressaltar que, no guia, em cada medida são descritos também o seu objetivo e as questões relacionadas.

**Tabela 9: Medidas**

<b>ID</b>	<b>Nome da Medida</b>	<b>Descrição</b>	<b>Função de Medição</b>	<b>Interpretação</b>	<b>Método de Coleta</b>	<b>Ref.</b>	<b>Subcaracterística Associada</b>
<b>M 01</b>	Grau de Disponibilidade (GD)	Capacidade de prestar serviço quando requisitado	$X = A$ , onde $A$ é $\frac{TmF}{TmF + TmR}$ $TmF$ = Tempo médio entre falhas $TmR$ = Tempo médio para reparar	Quanto mais próximo de 1, melhor	Automática	Retirada de [Kryvinska; Strauss e Zintherhof, 2011]	Disponibilidade
<b>M 02</b>	Taxa de Erros (IE)	Taxa de ocorrência de erros durante o uso do serviço	$X = A$ , onde $A$ é a quantidade de erros/tempo de uso do serviço	Quanto mais próximo de 0, melhor	Automática	Retirada de [Lee e Yun 2008]	Disponibilidade
<b>M 03</b>	Grau de Precisão (GP)	Representa a fração de contextos sentidos ou inferidos que são relevantes para o sistema	$\frac{P}{P + FP}$ $P$ = Positivos $FP$ = Falsos Positivos	Quanto mais próximo de 1, melhor	Automática	Adaptada de [Noury 2007]	Confiabilidade

<b>M 04</b>	Grau de Recall (abrangência ou cobertura) (GR)	Representa a fração de contextos relevantes para o sistema que foram corretamente sentidos ou inferidos	$\frac{P}{P + FN}$ P = Positivos FN = Falsos Negativos	Quanto mais próximo de 1, melhor	Automática	Adaptada de [Noury 2007]	Confiabilidade
<b>M 05</b>	Grau de Especificidade (GE)	Representa a fração de contextos não relevantes identificados pelo sistema sobre o total de contextos não relevantes	$\frac{N}{N + FP}$ N = Negativos FP = Falsos Positivos	Quanto mais próximo de 1, melhor	Automática	Adaptada de [Noury 2007]	Confiabilidade
<b>M 06</b>	Grau de Desempenho <i>F-Measure</i> (GDF)	Representa a média ponderada entre o grau de precisão e o grau de recall	$\frac{2 * (\frac{P}{P + FP} * \frac{P}{P + FN})}{\frac{P}{P + FP} + \frac{P}{P + FN}}$ P = Positivos FN = Falsos Negativos FP = Falsos Positivos	Quanto mais próximo de 1, melhor	Automática	Adaptada de [Noury 2007]	Confiabilidade
<b>M 07</b>	Grau de Correção ou Acurácia (GC)	Representa a fração de contextos corretamente sentidos ou inferidos	X = A, onde A é $\frac{P + N}{P + N + FN + FP}$ FP = Falsos positivos	Quanto mais próximo de 1, melhor	Automática	Adaptada de [Ranganathan 2005]	Confiabilidade

			P = Positivos FN = Falsos Negativos N = Negativos				
<b>M 08</b>	Tempo de Adaptação (TA)	Representa o tempo necessário para que a adaptação ocorra, buscando identificar inicializações precoces ou tardias do sistema.	$X = T$ e $Dp$ , onde $T$ é a média dos tempos que o sistema demorou em se adaptar uma vez que mudou o contexto e $Dp$ é o desvio padrão desses tempos	Alto: minutos Médio: segundos Baixo: milissegundos	Automática	Retirada de [Santos 2013]	Disponibilidade
<b>M 09</b>	Grau de Adaptação (GA)	Busca identificar se o sistema se adaptada sempre que necessário. Implicando na ausência de notificações equivocadas ou provimento de serviço baseado em informações atualizadas.	$X = \frac{(\sum_{j=1}^N \frac{A_j}{B_j}) * 100}{N}$ $A_j = n^\circ$ de vezes que adaptou $B_j = n^\circ$ de vezes que foi requisitada a adaptação mudança contextual) $N = n^\circ$ dos tipos	Quanto mais próximo de 100%, melhor.	Automática	Retirada de [Santos 2013]	Confiabilidade

das diferentes adaptações							
<b>M 10</b>	Grau de Corretude na Execução da Adaptação (GCA)	Representa a execução correta dos mecanismos ou regras de adaptação implicando em um provimento de serviço conforme o esperado.	$X = A$ , onde A é (1) Alto (2) Médio (3) Baixo	(1) Alto: execução correta das regras ou mecanismos de adaptação.  (2) Médio: execução de mais de uma regra ou mecanismo de adaptação, implicando em uma adaptação incorreta. Porém não inviabilizando o provimento do serviço.  (3) Baixo: execução de mais de uma regra ou mecanismo de adaptação sobre a mesma estrutura, implicando em uma adaptação incorreta e inviabilizando o provimento do serviço.	Análise do Desenvol-vedor	Baseada em [Cheng 2009]	Confiabilidade
<b>M 11</b>	Grau de Discrição de mecanismos de	Tempo usado para interagir com o subsistema de segurança (por exemplo,	$X = A$ , onde A é (1) Alto	(1) Alto: minutos (2) Médio: segundos	Observação	Adaptada de [Ranganath	Disponibilidade

	segurança (GDM)	autenticação) para auxiliar a tarefa principal.	(2) Médio (3) Baixo	(3) Baixo: milissegundos		an 2005]	
<b>M 12</b>	Grau de Compreensão do Uso de Dados (GCUD)	Mede a capacidade do usuário de entender quais dos seus dados ou informações pessoais são utilizados pelo sistema.	<b>X = A</b> , onde A é (1)Alto (2)Médio (3)Baixo	(1) Alto: o usuário é ciente de todas as suas informações que são utilizadas pelo sistema. (2) Médio: o usuário é ciente de apenas algumas das suas informações que são utilizadas pelo sistema. (3) Baixo: o usuário não é ciente de nenhuma das suas informações que são utilizadas pelo sistema.	Questionário do Usuário	Baseada em [Scholtz e Consolvo 2004]	Proteção
<b>M 13</b>	Grau de Permissão no Uso de Dados (GPUD)	Representa a existência de recursos disponíveis ao usuário para controlar as permissões de uso sobre suas informações pessoais: privacidade do conteúdo, identidade, localização e outros.	<b>X = A</b> , onde A é (1)Alto (2)Médio (3)Baixo	(1) Alto: o usuário tem controle sobre todos os seus dados que são utilizados pelo sistema e pode alterar as permissões a qualquer momento. (2) Médio: existem recursos disponíveis para o usuário controlar e alterar a permissão de alguns dos seus dados pessoais que são utilizados pelo sistema. (3) Baixo: Não existem recursos disponíveis para o usuário controlar	Análise do Usuário	Baseada em [Scholtz e Consolvo 2004]	Confidencialidad e

ousos dos seus dados por parte do sistema.

<b>M 14</b>	Grau de Concordância das Ações do Sistema (GCAS)	O número de funções providas pelo sistema deve ser compatível com o número de funções executadas e esperadas (e.g., ao entrar em um ambiente o usuário espera receber uma única informação, logo, o sistema não deve fornecer nenhuma ou mais de uma informação).	$X = A/B$ Onde, A = Número de funções executadas e esperadas B = Número de funções providas pelo serviço ubíquo	Quanto mais próximo de 1, melhor	Automático e Análise do Usuário	Adaptada de [Lee e Yun 2008]	Confiabilidade
<b>M 15</b>	Grau de independência ou autonomia do sistema (GIA)	Mede a necessidade de requisições de ajuda, por parte do sistema, para confirmar o resultado de alguma saída e/ou serviço.	$X = A$ , onde A é (1)Alto (2)Médio (3)Baixo	(1)Alto: o sistema não necessita ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço. (2)Médio: o sistema necessita da ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço, porém as requisições de ajuda estão dentro do previsto na especificação do sistema e fazem parte de um conjunto de	Automático e Análise do Usuário	Adaptada de [Scholtz e Consolvo 2004]	Confiabilidade

				requisições de ajuda previamente esperadas/configuradas. (3)Baixo: o sistema necessita da ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço em situações onde esta requisição de ajuda não está previamente configurada/esperada.			
<b>M 16</b>	Presença de Políticas de Segurança (PPS)	Representa a existência de políticas de segurança utilizadas pelo sistema ou pela aplicação	<b>X = A</b> , onde A é (0) Nenhuma (1) Presença de Política de Concessão Baseada em Permissão (2) Presença de Política de Interação	(0) Nenhuma (1) Presença de Política de Concessão Baseada em Permissão (2) Presença de Política de Interação	Automático e Análise do Desenvolvedor	Baseada em [Kilinc et al 2012]	Proteção
<b>M 17</b>	Grau de Documentação (GDoc)	Avaliar a existência de documentação associada ao sistema desenvolvido	<b>X = A</b> , onde A é (0) Alto (1) Médio (2) Baixo	(0) Alto: Todos os componentes estão documentados. (1) Médio: Apenas uma parte dos componentes estão documentados. (2) Baixo: Não há documentação.	Análise do Desenvolvedor	Indicada em Reunião GQM	Manutenibilidade

<b>M 18</b>	Grau de Esforço para Realizar Manutenção do Sistema (GEMS)	Avaliar o esforço, relacionando-o ao tempo, necessário para realizar manutenção do sistema	$X = A$ , onde A é (0) Alto (1) Médio (2) Baixo	(0) Alto: O tempo para realizar manutenção é superior ao tempo que foi necessário para implantação. (1) Média: O tempo para realizar manutenção é equivalente ao tempo que foi necessário para implantação. (2) Baixo: O tempo para realizar manutenção é inferior ao tempo que foi necessário para implantação.	Análise do Desenvolve- dor	Indicada em Reunião GQM	Manutenibilidade
<b>M 19</b>	Grau de Esforço para Realizar Manutenção ou Evolução de Código (GEEC)	Avaliar o esforço, relacionando-o ao tempo, necessário para realizar manutenção ou evolução do código	$X = A$ , onde A é (0) Alto (1) Médio (2) Baixo	(0) Alto: O tempo para realizar manutenção ou evolução é superior ao tempo que foi necessário para codificação. (1) Média: O tempo para realizar manutenção ou evolução é equivalente ao tempo que foi necessário para codificação. (2) Baixo: O tempo para realizar manutenção ou evolução é inferior ao tempo que foi necessário para codificação.	Análise do Desenvolve- dor	Indicada em Reunião GQM	Manutenibilidade

<b>M 20</b>	Grau de Integridade dos Dados Transmitidos em Rede (GIR)	Avaliar a existência da implementação de técnicas de integridade sobre os dados transmitidos em rede	$\mathbf{X} = \mathbf{A}$ , onde A é (0) Alto (1) Médio (2) Baixo	(0) Alto: Implementa técnicas de integridade com verificação e correção. (1) Média: Implementa técnicas de integridade, porém não garante verificação e correção. (2) Baixo: Não implementa técnicas de integridade.	Análise do Desenvolvedor	Indicada em Reunião GQM	Integridade
<b>M 21</b>	Grau de Integridade dos Dados Armazenados (GIA)	Avaliar a existência da implementação de técnicas de integridade sobre os dados armazenados	$\mathbf{X} = \mathbf{A}$ , onde A é (0) Alto (1) Médio (2) Baixo	(0) Alto: Implementa técnicas de integridade com verificação e correção. (1) Média: Implementa técnicas de integridade, porém não garante verificação e correção. (2) Baixo: Não implementa técnicas de integridade.	Análise do Desenvolvedor	Indicada em Reunião GQM	Integridade
<b>M 22</b>	Grau de Funcionamento em Background (GFB)	Avaliar se o sistema funciona em background	$\mathbf{X} = \mathbf{A}$ , onde A é (1) Alto (2) Médio (3) Baixo	(1) Alto: o sistema funciona normalmente em background (2) Médio: Algum aspecto do sistema é comprometido quando o dispositivo está bloqueado e a aplicação rodando, consequentemente, em background (3) Baixo: O sistema não funciona em		Indicada em Reunião GQM	Confiabilidade

<b>M 23</b>	Grau de Independência da Bateria (GI <sub>dB</sub> )	Avaliar se o sistema funciona sempre da mesma forma, mesmo com o dispositivo em diferentes níveis de bateria	X = A, onde A é (1) Alto (2) Médio (3) Baixo	(1) Alto: o sistema funciona conforme o especificado e sempre que necessário em qualquer nível de bateria  (2) Médio: Algum aspecto do sistema é comprometido quando o dispositivo está com pouca bateria (menos de 25%)  (3) Baixo: O sistema não funciona quando o dispositivo está com pouca bateria.	Indicada em Reunião GQM	Disponibilidade
<b>M 24</b>	Grau de Influência sobre a Bateria (GI <sub>nB</sub> )	Avaliar a influência do sistema sobre o consumo de bateria do dispositivo.	X = A, onde A é (1) Alto (2) Médio (3) Baixo	1) Baixo: a diferença entre o consumo de bateria com o sistema rodando ou sem rodar não ultrapassa 10%  (2) Médio: a diferença entre o consumo de bateria com o sistema rodando ou sem rodar não ultrapassa 15%  (3) Alto: a diferença entre o consumo	Indicada em Reunião GQM	Confiabilidade

de bateria com o sistema rodando ou sem rodar ultrapassa 15%.

<b>M 25</b>	Grau de Independência de Componentes Externos (GInCE)	Avaliar se o sistema funciona conforme o especificado e sempre que requisitado, independente de componentes externos ao dispositivo no qual ele executa.	X = A, onde A é (1) Alto (2) Médio (3) Baixo	<p>(1) Alto: O sistema funciona conforme o especificado e sempre que necessário mesmo que qualquer componente externo tenha algum problema.</p> <p>(2) Médio: Algum aspecto do sistema é comprometido quando algum componente externo, facilmente identificável, tem algum problema.</p> <p>(3) Baixo: O sistema não funciona quando qualquer componente externo tem algum problema.</p>	Indicada em Reunião GQM	Disponibilidade
-------------	---	--	---	--	-------------------------	-----------------

Onde *retirada* indica que a medida foi extraída do trabalho referenciado onde também é apresentada como uma medida de qualidade. *Adaptada* indica que a medida foi extraída de um trabalho onde é apresentada como uma possível medida, porém, não formatada seguindo algum padrão específico ou, não definida para o domínio em questão neste trabalho. *Baseada* indica que a medida foi extraída de um trabalho onde apenas existem conceitos que podem vir a formalizar uma possível medida, segundo os autores e, por fim, *indicada* em reunião GQM indica que a medida foi sugestão dos especialistas envolvidos no GQM, usando como base seus conhecimentos e experiências na área.

A medida M01 busca avaliar a disponibilidade do sistema, identificando a capacidade do sistema em prover serviços sempre que requisitado. Para isso, essa avaliação leva em consideração o tempo médio entre as falhas do sistema e o tempo médio para reparar as falhas.

A medida M02 busca avaliar a confiança com base na quantidade de erros que acontecem durante o uso do sistema. Para melhores resultados, em relação a confiança no funcionamento, os valores esperados para essa medida são sempre o mais próximo possível de zero, indicando assim a baixa ocorrência de erros.

A medida M03 busca avaliar a precisão do sistema, isto é, a capacidade do sistema de identificar que houve uma mudança específica de contexto onde, essa mudança é de fato importante para o provimento do serviço. Essa capacidade também é expressa em forma de porcentagem, quanto mais próximo de 100% maior a precisão do sistema e maior o impacto positivo da confiabilidade sobre a confiança no funcionamento do sistema.

Onde:

	Informação contextual visível ao sistema e importante (relevante) para o provimento do serviço	Informação contextual visível ao sistema, porém, não necessária (não relevante) para o provimento do serviço
Percebida pelo sistema como algo importante para o provimento do serviço	P (positivo)	FP (falso positivo)
Percebida pelo sistema como algo não importante para o provimento do serviço	FN (falso negativo)	N (negativo)

As definições expressas na tabela acima para P, FP, N e FN são válidas para todas as medidas nas quais esses critérios aparecem, bem como o significado de relevante e não relevante, mencionado na descrição das medidas e associados na tabela acima às informações contextuais utilizadas pelo sistema.

A medida M04 busca avaliar a cobertura ou abrangência dos contextos sentidos ou inferidos pelo sistema, isto é, a capacidade do sistema de identificar que houve uma mudança de contexto relevante para o provimento de serviço. Essa capacidade é expressa em forma de porcentagem, quanto mais próximo de 100% (ou 1) maior a cobertura ou abrangência do sistema e maior o impacto positivo da confiabilidade sobre a confiança do sistema. A medida representa a razão entre as mudanças que houve e o sistema detectou e todas as mudanças de contexto detectadas pelo sistema, sejam elas relevantes ou não. A interpretação para P e FP é a mesma utilizada na medida M03. Além disso, FN indica as mudanças que houveram e o sistema não foi capaz de detectar. O critério FN faz parte também da função de medição de outras medidas, possuindo o mesmo significado descrito em M4.

A medida M05 representa a fração de contextos não relevantes identificados pelo sistema sobre o total de contextos não relevantes, a idéia-chave dessa medida é que as mudanças de contexto que não são relevantes para o provimento de serviços e o total de mudanças não relevantes identificados seja compatível, conseqüentemente, quanto mais próximo de 1 melhor pois, isto implica em um baixo número de falsos positivos.

A medida M06 representa a média ponderada entre o grau de precisão e o grau de recall. Em estatística a média é o valor que aponta para onde mais se concentram os dados de uma distribuição. Pode ser considerada o ponto de equilíbrio. Na média simples todos os valores possuem um mesmo peso, situação diferente na média ponderada, que para cada valor deve-se levar em conta o valor do seu peso.

A medida M07 representa a porcentagem de mudanças contextuais corretamente sentidas ou deduzidas. Para isso é avaliado a razão entre a soma de positivos e negativos (representando contextos relevantes) e a soma de falsos positivos, falsos negativos, positivos e negativos (representando todos os contextos).

Diferentemente de M03, a medida M07 pode não ser útil (e.g., gerar valores de interpretação próximos do melhor possível para sistemas onde a necessidade de melhorias é evidente) em alguns cenários (e.g., valores altos para N e valores baixos para P, FN e FP, podem gerar um grau de acurácia alto para um sistema onde o grau de precisão é baixo).

A medida M08 mede o tempo que o sistema demora para se adaptar a partir do instante em que a informação de contexto foi sentida ou inferida. 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 a média entre os tempos de adaptação e o desvio-padrão (visto que é necessário verificar se houve resultados muito distantes uns dos outros).

A medida M09 busca avaliar se as adaptações ocorreram no momento esperado, mesmo que de maneira errada. Essa medida foi definida pois o sistema pode não conseguir se adaptar, em alguns casos, mesmo diante de uma mudança contextual. Para calcular essa medida, é necessário identificar quais adaptações e mudanças de contexto o sistema possui e que podem ser coletadas e, então, 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 medida M10 busca avaliar aspectos do sistema que possibilitem garantir que o serviço é entregue ao usuário de forma adequada.

A medida M11 busca avaliar o tempo que o sistema precisa para interagir com os mecanismos de segurança existentes. Por exemplo, no momento do *login* o sistema leva tempo longo para autenticar o usuário, isso afeta a confiabilidade e disponibilidade do sistema, pois a prestação de serviço quando requisitado demora e o serviço não é prestado conforme especificado.

A medida M12 busca avaliar se há ou não recursos disponíveis ao usuário para entender e compreender como os dados gravados são utilizados. Por exemplo, em sistemas que utilizam informações sobre a localização do usuário, o usuário é ciente desta utilização? Quanto maior o número de recursos disponíveis mais próximo de 'Alto' o valor da medida, quanto menor, mais próximo de 'Baixo'.

A medida M13 busca avaliar se há ou não recursos disponíveis ao usuário para controle sobre suas informações pessoais: privacidade do conteúdo, identidade, localização e outros. Por exemplo, é possível alterar permissões? É possível restringir conteúdo? Quanto maior o número de recursos disponíveis mais próximo de 'Alto' o valor da medida, quanto menor, mais próximo de 'Baixo'.

A medida M14 busca avaliar se o número de funções providas pelo sistema e o número de funções esperadas é compatível. Para isso é utilizada a razão entre o número de funções providas e o número de funções requisitadas. Quanto mais próximo de 1, melhor.

A medida M15 busca avaliar a independência do sistema para tomar decisões, isto é, a necessidade de requisições de ajuda, por parte do sistema, para confirmar o resultado de alguma saída e/ou serviço. Quanto maior o número de requisições de ajuda mais próximo de ‘Alto’ o valor da medida, quanto menor, mais próximo de ‘Baixo’.

A medida M16 busca avaliar a presença de políticas de segurança utilizadas pelo sistema ou pela aplicação. De acordo com [Kilinc et al 2012], referência da qual foi retiradas informações para definição desta medida, as políticas de segurança em sistemas ubíquos e são: Políticas de Concessão Baseada em Permissão e Políticas de Interação.

A medida M17 tem como objetivo arantir que o sistema desenvolvido possui documentação, auxiliando no seu entendimento e utilização.

As medidas M18 e M19, por sua vez, buscam garantir que o tempo necessário para a manutenção ou evolução do sistema e do código sejam adequadas. Não implicando em perdas ou atrasos para o provimento do serviço conforme o especificado.

As medidas M20 e M21 objetivam avaliar e garantir que os dados utilizados pelo sistema, que são transmitidos via rede ou armazenados, não sejam acessados por quem não tem autorização.

A medida M22 busca avaliar se o serviço é entregue ao usuário conforme o especificado e sempre que requisitado, mesmo quando o dispositivo estiver bloqueado (em descanso). Essa medida retrata aspectos específicos dos sistemas desenvolvidos para dispositivos móveis.

A medida M23 tem como objetivo garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado, mesmo quando o dispositivo estiver com bateria baixa. A ideia-chave desta medida é avaliar como o sistema se comporta quando a bateria do dispositivo está baixa (menos de 25%). Algumas funções nativas do desenvolvimento para sistemas *Android* (o comando *broadcast*, por exemplo) não são ativadas (ou ativadas em intervalos de tempo maiores do que o adequado) quando o dispositivo está com pouca bateria.

A medida M24 busca garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado sem comprometer significativamente (e.g., 15% a mais de consumo quando utilizando o sistema) a bateria do dispositivo.

A medida M25 objetiva garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado independente de componentes externos. Um exemplo de questionamento que essa medida pode ajudar a sanar é: em um sistema que coloca o celular no mudo de acordo com os compromissos da *Google* agenda do usuário. Caso ocorra algum problema de indisponibilidade com o sistema de agendas da *google*, o sistema presente no dispositivo, que interage com essa agenda, também deixará de funcionar?

### 4.3 Coleta das Medidas

Como mencionado no Capítulo 2, essa fase do GQM consiste em coletar, armazenar e processar todas as informações necessárias para calcular as medidas (SANTOS 2014). Definindo, portanto, como é a obtenção dos dados que são utilizados na medição. De acordo com (VAN SOLINGEN E BERGHOUT, 1999), durante a fase de coleta de dados, é importante observar quatro 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?

De acordo com (ROCHA et al 2012), como a produção de software é uma tarefa intelectual, a coleta dos dados requer, muitas vezes, observação e coleta manual. Assim, segundo o autor, para garantir que a coleta seja feita de forma correta, é necessário definir como ela é realizada.

Neste trabalho, para coletar as medidas estabelecidas, é necessário utilizar três diferentes métodos:

- Coleta Automática: a coleta de dados automática é feita com base em dados obtidos durante o uso da aplicação (em testes) e armazenados em *log*.
- Análise do Usuário: a análise com o usuário é feita por meio de formulários e questionários e procura avaliar a confiança do usuário durante a utilização do sistema.
- Análise do Desenvolvedor: a análise com o desenvolvedor é feita por meio de formulários e questionários e busca auxiliar o desenvolvedor na construção de um sistema no qual o usuário possa justificadamente confiar.

Para cada medida definida, o método de coleta utilizado é apresentado no Apêndice A, no tópico *Procedimento de Medição*. Além disso, outros aspectos relacionados à coleta são apresentados nos tópicos *Momento da Medição*, *Periodicidade da Medição* e *Responsável pela Medição*.

### 4.4 Interpretação dos Resultados

Esta fase corresponde a quarta e última etapa da metodologia GQM. Como indicado por (VAN SOLINGEN; BERGHOUT, 1999), o objetivo nesse estágio é analisar os resultados obtidos na coleta dos dados e utilizá-los para interpretar as questões e os objetivos de medição definidos.

Como apresentado na Seção 4.3, os valores de interpretação ou, intervalo esperado dos dados, representam aproximações do que seria um bom resultado. Pois, como apontado por (SANTOS 2014), valores de interpretação específicos não são aplicáveis, pois este resultado depende do propósito do sistema, o que varia de aplicação para aplicação. Além disso, não existem valores históricos com os quais podemos comparar e especificar uma escala de interpretação mais específica.

Para cada medida definida, o procedimento de análise é detalhado no Apêndice A, nos tópicos *Momento da Análise*, *Periodicidade da Análise* e *Responsável pela Análise*.

## 4.5 Considerações Finais

Neste capítulo foi apresentado um conjunto de vinte e cinco medidas de qualidade de *software* para avaliação da confiança no funcionamento em sistemas ubíquos que executam em dispositivos móveis. Esse conjunto de medidas foi definido por meio da metodologia *Goal-Question-Metric* (GQM).

Esta seção busca apresentar as discussões sobre as medidas definidas abordando a adequação e cobertura destas para os sistemas ubíquos para dispositivos móveis. Para tanto, é evidenciada a relação entre as medidas e as possíveis causas para falhas nesse tipo de sistema.

Esta relação foi priorizada pois, como apresentado no Capítulo 2, a ocorrência destas é capaz de afetar a confiança que o usuário possui no funcionamento do sistema. Portanto, apresentando esse relacionamento, é possível mostrar que com a utilização das medidas definidas pode-se minimizar a ocorrência de falhas e, por consequência, fortalecer a confiança no funcionamento.

Para isso, será utilizada a classificação encontrada no estudo de (ROCHA 2010), apresentada em resumo, a seguir:

- Falhas de Dispositivo: Sistemas ubíquos podem ser compostos por diferentes tipos de dispositivos (e.g., computadores de mesa, notebooks, sensores, *smartphones* e outros). Cada dispositivo é suscetível a um conjunto particular de faltas que, em maior ou menor grau, podem contribuir para a falha do dispositivo e do sistema como um todo. Para minimizar esse tipo de falha o desenvolvedor deve estar atento a questões relativas ao uso da bateria, a corretude dos sensores utilizados e outros.
- Falhas de Rede: Os componentes que fazem parte de um sistema ubíquo podem estar conectados através de enlaces de redes, com ou sem fio. Diante desse cenário é preciso projetar e desenvolver sistemas que saibam lidar com problemas como, por exemplo, a baixa

força do sinal ou a indisponibilidade de canais de comunicação devido a um alto tráfego na rede.

- Falhas de Aplicação: As falhas de aplicações podem ser causadas por faltas de código (*bug*), erros do sistema operacional, exceções não tratadas e mau uso por parte do usuário. Além disso, as aplicações podem ser vítimas de softwares maliciosos (e.g., vírus e *worms*).
- Falhas de Serviço: Os ambientes ubíquos são repletos de serviços. Esses serviços são responsáveis por prover diferentes funcionalidades para as aplicações e para o próprio sistema, sendo susceptíveis aos mesmos tipos de falhas das aplicações, apresentadas anteriormente.
- Falhas de Adaptação: Em sistemas ubíquos, o subsistema de adaptação é responsável por ajustar tanto os serviços como as aplicações às mudanças de contexto. Durante o processo de adaptação, faltas podem fazer com que a estrutura das aplicações e/ou dos serviços seja modificada de maneira errada. Isso faz com que as aplicações e/ou serviços passem a gerar um comportamento inesperado e, possivelmente, errôneo. Isso pode ocorrer tanto pela especificação errada da regra de adaptação quanto pelo conflito entre regras.
- Falhas de Contexto: O subsistema de contexto é responsável pela gerência do contexto desde a fase de aquisição até a notificação aos demais componentes do sistema. Falhas no subsistema de contexto podem ocasionar a inicialização precoce ou tardia de aplicações, adaptações indevidas ou inferências errôneas.

A Figura 15 sumariza a relação entre as medidas definidas e as causas das falhas em sistemas ubíquos. Na parte superior, estão as falhas em sistemas ubíquos. Na parte lateral esquerda da figura, são descritas as medidas. Os quadrados escurecidos indicam que a medida da linha horizontal está relacionada com a falha da linha vertical e pode ser utilizada para minimizar (direta ou indiretamente) a ocorrência da falha.

	<i>Falha de Aplicação</i>	<i>Falha de Dispositivo</i>	<i>Falha de Serviço</i>	<i>Falha de Adaptação</i>	<i>Falha de Contexto</i>	<i>Falha de Rede</i>
M 01						
M 02						
M 03						
M 04						
M 05						
M 05						
M 07						
M 08						
M 09						
M 10						
M 11						
M 12						
M 13						
M 14						
M 15						
M 16						
M 17						
M 18						
M 19						
M 20						
M 21						
M 22						
M 23						
M 24						
M 25						

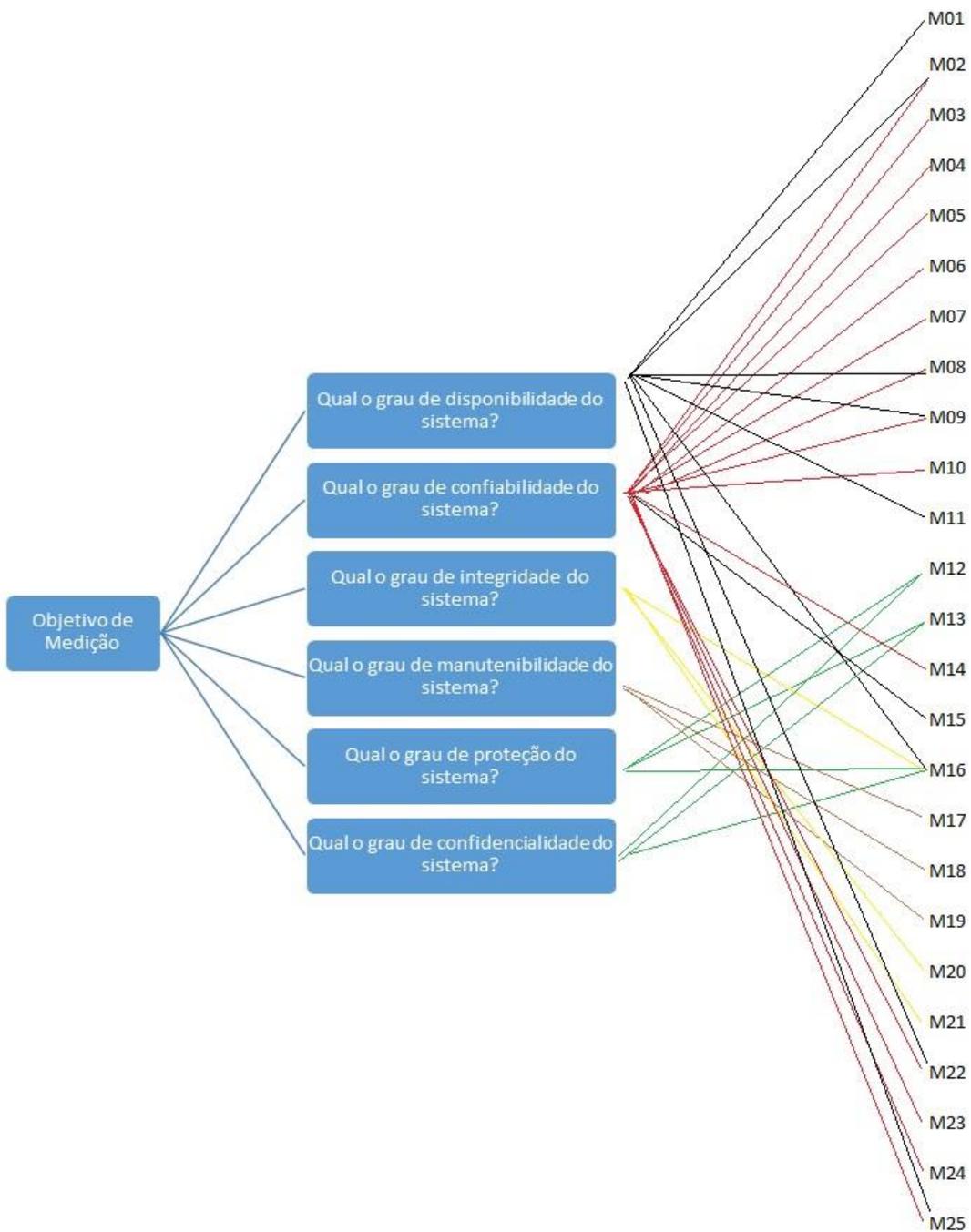
**Figura 15: Relação entre medidas e falhas**

Como pode-se observar na Figura 3, as medidas M01, M02, M23 e M24 podem minimizar falhas de dispositivo pois estão diretamente associadas a avaliação da capacidade do sistema de prover serviços sempre que requisitado, sendo, portanto, impactadas por questões associadas ao uso da bateria, a corretude dos sensores utilizados e outros. Além disso, a baixa força do sinal ou a indisponibilidade de canais de comunicação também podem ser identificados por meio da coleta dessas medidas, assim, incluindo também a medida M10 e M20, é possível minimizar falhas de rede.

As medidas M01 – M25 são capazes de auxiliar a identificação de falhas que podem ser causadas por faltas de código (*bug*), erros do sistema operacional, exceções não tratadas e mau uso por parte do usuário, sendo assim é possível minimizar falhas de aplicação e serviço.

Por fim, as medidas M02 – M10, M13 – M15 e M20 – M25 podem minimizar falhas de contexto e adaptação pois procuram avaliar questões relativas ao contexto desde a fase de aquisição até a notificação dos demais componentes do sistema e como se ajustam de acordo às mudanças de contextuais.

Como podemos observar, no total foram apresentadas 25 medidas. O modelo final do GQM, seguindo a mesma representação visual de (SANTOS 2014), é apresentado na Figura 16:



**Figura 16: Modelo Final GQM**

# Capítulo 5

## Estudos de Caso

Este capítulo apresenta, por meio de estudos de caso, a aplicação das medidas definidas no capítulo anterior. 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 *GREat Mute*. A Seção 5.3 apresenta a coleta e interpretação das medidas na aplicação *GREat Tour*, a Seção 5.4 apresenta a coleta e interpretação das medidas na aplicação *fAlert* e, por fim, a Seção 5.5 apresenta como considerações finais do capítulo, uma reflexão sobre a utilidade das medidas quando coletadas em sistemas ubíquos.

### 5.1 Planejamento

Com o intuito de avaliar o uso das medidas apresentadas no Capítulo 4, estudos de casos (EC) são utilizados. O estudo de caso trata-se de uma abordagem de investigação, especialmente adequada quando procuramos compreender, explorar ou descrever acontecimentos e contextos, nos quais estão simultaneamente envolvidos diversos fatores (YIN 1994).

Por ter várias aplicações, os estudos de caso são apropriados para pesquisadores individuais, pois dão a oportunidade para que um aspecto de um problema seja estudado em profundidade dentro de um período de tempo limitado. Além disso, dado a sua flexibilidade, são úteis na exploração de novos processos, pois, auxiliam a compreensão dos novos comportamentos e concepções.

Esse método de investigação foi escolhido, pois:

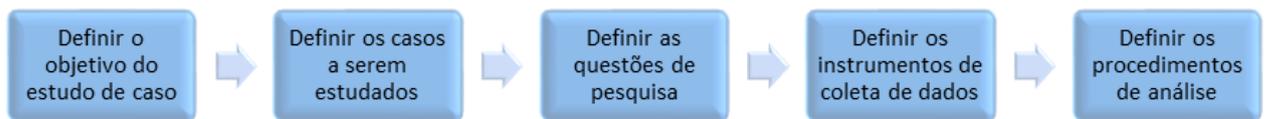
- É utilizado por projetos que utilizam a abordagem GQM, na fase de coleta de dados (VAN SOLINGEN e BERGHOUT 1999);
- Permite a coleta de dados tanto quantitativos, quanto qualitativos; e
- É conduzido em configurações do mundo real e, assim, tem um alto grau de realismo, permitindo, portanto, realizar testes com usuários interagindo com um sistema real.

Os estudos de caso se caracterizam por (YIN 1994):

- **Objetivar a descoberta:** o investigador se mantém atento a novos elementos que podem surgir, buscando novas respostas e novas indagações no desenvolvimento do seu trabalho.
- **Enfatizar a interpretação contextual:** para melhor compreender um problema, deve-se relacionar as ações, os comportamentos e as interações das entidades envolvidas com a problemática da situação.
- **Retratar a realidade de forma completa e profunda:** o pesquisador enfatiza a complexidade da situação, procurando revelar a multiplicidade de fatos que a envolvem e a determinam.
- **Utilizar várias fontes de informação:** o pesquisador recorre a uma variedade de dados, coletados em diferentes momentos, em situações variadas e com uma variedade de tipos de informantes.

Como mencionado anteriormente, para a realização dos estudos de caso são utilizadas três aplicações ubíquas para dispositivos móveis, com o intuito de avaliar a confiança no funcionamento nesse tipo de sistema. Os resultados obtidos são apresentados nas seções a seguir.

Para o planejamento dos estudos de caso, deste trabalho foram seguidas as mesmas atividades de planejamento propostas por (RUNESON; HÖST, 2008) e utilizadas por (SANTOS 2014), apresentada na Figura 17:



**Figura 17: Planejamento para os estudos de caso**

O objetivo do estudo de caso é baseado na motivação desta pesquisa, avaliar a confiança no funcionamento em sistemas ubíquos. Como casos a serem estudados, estão as três aplicações ubíquas mencionadas no início deste capítulo. As questões de pesquisa são as mesmas derivadas no GQM e apresentadas no Capítulo 4. Por fim, os instrumentos de coleta e procedimentos de análise variam de acordo com cada medida e podem ser encontrados no guia apresentado no Anexo A.

Para cada uma das aplicações, o seguinte grupo de pessoas está envolvido na realização dos estudos de caso (Tabela 10):

**Tabela 10: Perfil dos usuários envolvidos nos estudos de caso**

	<b>GREat Mute</b>	<b>fAlert</b>	<b>GREat Tour</b>
<b>Quantidade Total de Pessoas</b>	3	4	2
<b>Apenas Usuário</b>	1	3	0
<b>Desenvolvedor e Usuário</b>	2	1	2
<b>Média de Idade</b>	22 anos	23 anos	22 anos
<b>Escolaridade</b>	Graduandos	Graduandos	Graduandos
<b>Familiaridade com o uso de Smartphones</b>	Alta para todos	Alta para todos	Alta para todos
<b>Familiaridade com o uso de sistemas ubíquos</b>	Média para todos	Média para todos	Média para todos

Além disso, os seguintes dispositivos foram utilizados (Tabela 11):

**Tabela 11: Dispositivos usados nos estudos de caso**

<b>Samsung Galaxy S2 Slim Duos</b>	
<b>Dimensões</b>	133 x 66 x 9,7 mm
<b>Tamanho de tela</b>	4,5 polegadas 960x540 pixel
<b>Peso</b>	139g
<b>Tecnologias sem fio</b>	Wi-Fi, 3G ,EDGE e GPRS
<b>Armazenamento</b>	8 GB
<b>Memória RAM</b>	1 GB
<b>Processador</b>	Quad-core 1,1 Ghz
<b>Bateria</b>	Li-Lon 2100 mAh
<b>Qualidade do som</b>	

<b>Sensores</b>	Acelerômetro, microfone, proximidade, GPS, barômetro e Bússola
<b>Sistema Operacional</b>	Android 4.0.2 (Jelly Bean)

Samsung Galaxy S3 Slim Duos	
<b>Dimensões</b>	133 x 66 x 9,7 mm
<b>Tamanho de tela</b>	4,5 polegadas 960x540 pixel
<b>Peso</b>	139g
<b>Tecnologias sem fio</b>	Wi-Fi, 3G ,EDGE e GPRS
<b>Armazenamento</b>	16 GB
<b>Memória RAM</b>	1 GB
<b>Processador</b>	Quad-core 1,2 Ghz
<b>Bateria</b>	Li-Lon 2100 mAh
<b>Qualidade do som</b>	
<b>Sensores</b>	Acelerômetro, microfone, proximidade, GPS, barômetro e Bússola
<b>Sistema Operacional</b>	Android 4.2 (Jelly Bean Plus)

Motorola Moto G2	
<b>Dimensões</b>	141.5 x 70.7 x 6.0-11.0mm (curvo)
<b>Tamanho de tela</b>	5 polegadas HD 720x1280 pixel
<b>Peso</b>	149 gramas
<b>Tecnologias sem fio</b>	802.11 b/g/n, Bluetooth v 4.0,
<b>Armazenamento</b>	8 Gb
<b>Memória RAM</b>	1 Gb
<b>Processador</b>	quad-core 1,2 GHz
<b>Bateria</b>	2070 mAh
<b>Qualidade do som</b>	
<b>Sensores</b>	
<b>Sistema Operacional</b>	Android 4.4.4, KitKat®

Nexus 4	
<b>Dimensões</b>	133 x 68 x 9,1 mm
<b>Tamanho de tela</b>	4,7 Polegadas 768x1280
<b>Peso</b>	139,5 g
<b>Tecnologias sem fio</b>	Bluetooth, Wi-Fi a/b/n/g, 3G, EDGE e GPRS
<b>Armazenamento</b>	16 GB
<b>Memória RAM</b>	2 Gb
<b>Processador</b>	Qualcomm Snapdragon APQ8064 Quad-core 1.5 GHz
<b>Bateria</b>	2100 mAh

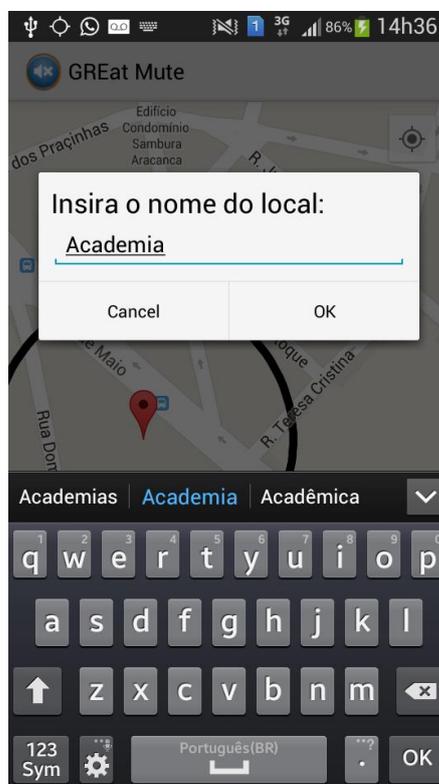
Qualidade do som	
Sensores	Acelerômetro, GPS, giroscópio e proximidade
Sistema Operacional	Android 4.2 (Jelly Bean Plus)

Os resultados obtidos são apresentados nas demais seções deste capítulo.

## 5.2 GREat Mute

A GREat Mute consiste em uma aplicação para *Smartphones* com sistema operacional *Android*, capaz de silenciar o dispositivo móvel do usuário em determinadas condições nas quais isso é adequado.

Executando em background, a aplicação monitora o *Google Calendar* do usuário. Ao encontrar eventos nos quais o usuário considere que não é adequado receber ligações, como “reunião”, “aula” ou “cinema” a aplicação verifica se o usuário encontra-se ou não no local associado ao determinado compromisso e, caso ele esteja no referido local, o dispositivo móvel aciona o modo silencioso.





**Figura 18: GREAt Mute Versão 2.0**

A aplicação encontra-se atualmente na versão 3.0. Anteriormente, na versão 1.0 a aplicação tinha acesso apenas ao *Google Calendar* do usuário e colocava o dispositivo móvel no silencioso baseada apenas nas informações obtidas na agenda. Caso o usuário não comparecesse ao local do evento no horário marcado, a aplicação não era capaz de identificar essa situação, gerando, por vezes, o provimento do serviço de forma incorreta ou indesejada.

A versão 2.0 utiliza o GPS para verificar a localização do usuário e, a versão 3.0 faz a combinação do endereço obtido do *Google Calendar* e a localização obtida com o GPS, sem que seja necessário o usuário determinar explicitamente a localização do seu compromisso no mapa.



**Figura 19: GREAt Mute Versão 1.0**

A coleta das medidas para a avaliação da confiança, realizada nas duas versões (2.0 e 3.0), é apresentada na Seção 5.1. Além disso, é explicado como a interpretação dos resultados da versão 1.0, obtidos do estudo de (SANTOS 2014), em conjunto com novos testes dessa versão, utilizando as medidas apresentadas nesta dissertação, serviram para obter informações relevantes sobre a qualidade do sistema e identificar pontos de melhoria, implantados nas versões 2.0 e 3.0.

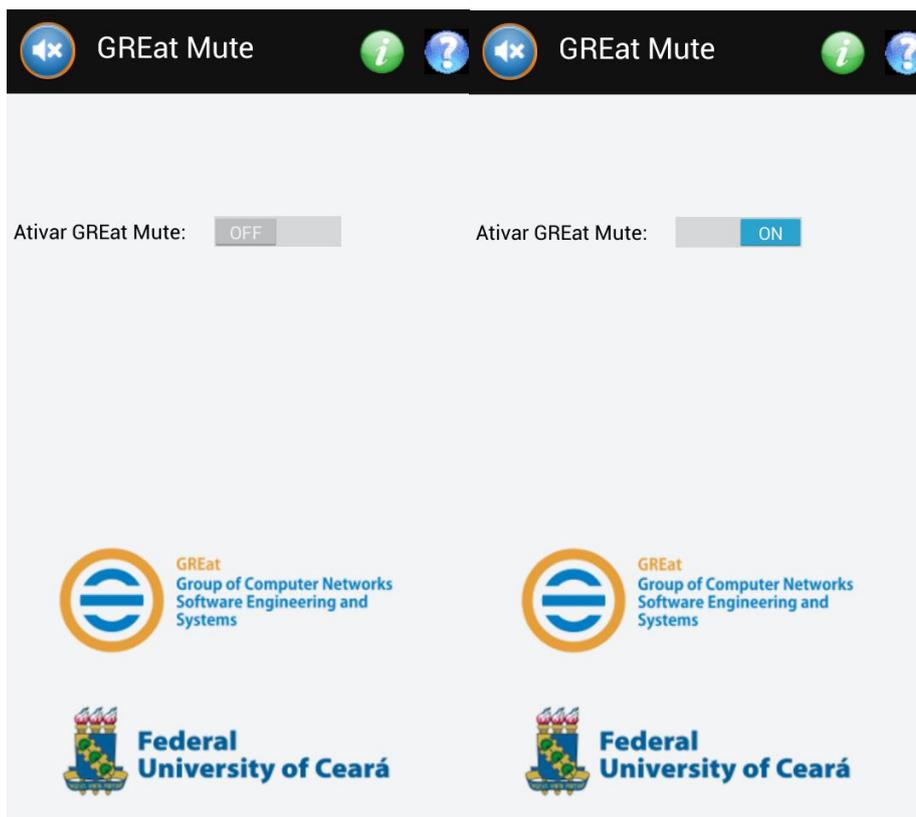


Figura 20: GREAt Mute Versão 3.0

### 5.2.1 GREAt Mute v 1.0

Como mencionado na Seção 5.2, a versão 1.0 da GREAt Mute utiliza apenas algumas (e.g., nome do compromisso, horário do compromisso) das informações obtidas do Google Calendar no usuário. Não fazendo uso de sensores ou informações que indiquem a real localização do usuário nos horários associados à compromissos.

Para a análise dessa versão foram consideradas informações obtidas das coletas realizadas por (SANTOS 2014) em seu trabalho, onde a aplicação é chamada de *UbiMute*. Além disso, foi necessário realizar novos testes com o aplicativo *GREAt Mute* 1.0 pois, as informações provenientes do estudo de (SANTOS 2014) não eram suficiente para a coleta de todas as medidas desta dissertação.

Foram realizados 150 testes com dois usuários, utilizando os dispositivos *Samsung Galaxy S3* e *Samsung Galaxy S4*. Nesta seção são apresentados os resultados obtidos pela coleta de dados na aplicação utilizando, para cada subcaracterística da confiança no funcionamento, uma questão do

GQM (apresentadas no Capítulo 4). Cada questão é comentada de acordo com os resultados das medidas obtidas.

1. Qual o grau de disponibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M 01</b>	Grau de Disponibilidade	X = 1
<b>M 02</b>	Taxa de Erros	X = 0
<b>M 08</b>	Tempo de Adaptação	X = T = 63 milissegundos Dp = 112 milissegundos
<b>M 11</b>	Discrição dos mecanismos de segurança	N/A
<b>M 23</b>	Grau de Independência da Bateria	X = 0 Alto
<b>M 24</b>	Grau de influência sobre a bateria	X = 1 Médio
<b>M 25</b>	Grau de independência de componentes externos	X = 1 Médio

O grau de disponibilidade do sistema em sua versão 1.0 pode ser considerado alto, apesar de existir dependência em relação a componentes externos (e.g., google agenda), o sistema não tem erros (M02) e está disponível sempre que requisitado (M01). Além disso, o tempo de adaptação é aceitável para os usuários que realizaram os testes e o comportamento do sistema, com relação a disponibilidade, não é afetado por índices baixos de bateria.

2. Qual o grau de integridade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M19</b>	Grau de Integridade dos Dados Transmitidos em Rede	X = 2 Baixo
<b>M20</b>	Grau de Integridade dos Dados Armazenados	X = 2 Baixo

Com relação ao grau de integridade, a versão 1.0 não implementa nenhum mecanismo para garantia da integridade dos dados transmitidos pelas diferentes partes envolvidas na

entrega do serviço. Também não há garantias de integridade para os dados armazenados no dispositivo. Logo, há necessidade de melhorias nessa sob o ponto de vista do desenvolvedor. Porém, sob o ponto de vista do usuário final, a falta de mecanismos de integridade não é identificável e, como não houveram problemas oriundos disto, não houve nenhum efeito sobre o grau de confiança do funcionamento do usuário no sistema.

### 3. Qual o grau de manutenibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M16</b>	Grau de Documentação	X = 0 Baixo
<b>M17</b>	Grau de Esforço para Realizar Manutenção do Sistema	X = 1 Médio
<b>M18</b>	Grau de Esforço para Realizar Manutenção ou Evolução do Código	X = 1 Médio

Com relação ao grau de manutenibilidade, a coleta das medidas indica a necessidade de documentação, pois a ausência desta implica em esforço extra para realizar a manutenção e evolução do sistema para as versões seguintes.

### 4. Qual o grau de confiabilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M03</b>	Grau de Precisão	X = 0,27 Ou 27%
<b>M04</b>	Grau de Recall	X = 0,88 Ou 88%
<b>M07</b>	Grau de Acurácia	X = 0,92 Ou 92%
<b>M10</b>	Corretude na Execução da Adaptação	X = 1 Médio
<b>M05</b>	Grau de Especificidade	X = 65%
<b>M06</b>	Grau de Desempenho	X = 78%
<b>M09</b>	Grau de Adaptação	X = 92%

<b>M14</b>	Grau de Concordância	X = 1
<b>M15</b>	Grau de Independência ou Autonomia	X = M Médio
<b>M22</b>	Grau de Funcionamento em Background	X = B Baixo

Com relação ao grau de confiabilidade, a versão 1.0 da aplicação GREat Mute não obteve bons resultados. A principal questão é a grande geração de falsos positivos. Decorrentes da falsa identificação, por parte da aplicação, de que o usuário está realmente cumprindo um compromisso cadastrado em sua *Google* agenda.

Além disso, a aplicação tem diversos problemas devido a falta de sincronização aplicação-agenda quando em *background* (e.g., quando o dispositivo se encontra em modo de descanso ou bloqueado).

5. Qual o grau de proteção do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M11</b>	Grau de Compreensão no Uso dos Dados	X = 0 Alto
<b>M12</b>	Grau de Permissão no Uso dos Dados	X = 2 Baixo
<b>M15</b>	Presença de Políticas de Segurança	X = 0 Nenhuma

Por fim, o grau de proteção do sistema é considerado baixo, pois de acordo com os três aspectos levados em consideração para a sua avaliação, apenas um obteve resultados satisfatórios. Não dar ao usuário possibilidade de alterar quais informações contextuais podem ser usadas pela aplicação e o momento de uso, causa no usuário final, de acordo com as pessoas que realizaram o teste e medidas coletadas, certo desconforto.

## 5.2.2 GREat Mute v 2.0

Segundo indica o estudo de (SANTOS 2014), a versão 1.0 da GREat Mute possui diversos erros de adaptação, decorrentes de mudanças contextuais. Que puderam ser confirmados com a avaliação feita e apresentada na Seção 5.2.1. Isso ocorre pois, como a aplicação não faz uso de informações relativas à localização do usuário, o contexto muitas vezes não corresponde a situação real do utilizador.

Além disso, os resultados obtidos com a coleta das medidas definidas neste trabalho, também indicam a necessidade de melhorias, pois para a maioria das medidas, os resultados obtidos estão muito abaixo (e.g., menos da metade) daquilo que seria o ideal, de acordo com os valores de interpretação estipulados na definição das medidas.

Diante desse cenário, a aplicação foi evoluída, fazendo uso em sua nova versão, de informações provenientes do GPS do dispositivo móvel do usuário. Nessa versão, é necessário que o usuário indique, utilizando um mapa fornecido pela própria aplicação, a localização do compromisso cadastrado na sua Google agenda.

Após a evolução da aplicação, novos testes foram realizados, utilizando os dispositivos *Samsung Galaxy S3* e *Samsung Galaxy S4*.

Assim como na seção anterior, os resultados obtidos com a coleta das medidas são apresentados de forma a servirem como respostas para as questões GQM. Cada questão é comentada de acordo com os resultados das medidas obtidas.

1. Qual o grau de disponibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M01</b>	Grau de Disponibilidade	X = 1
<b>M02</b>	Taxa de Erros	X = 0
<b>M08</b>	Tempo de Adaptação	X = T = 63 milissegundos Dp = 112 milissegundos
<b>M 11</b>	Discrição dos mecanismos de segurança	N/A
<b>M 23</b>	Grau de Independência da Bateria	X = 1 Médio
<b>M 24</b>	Grau de influência sobre a bateria	X = 1 Médio
<b>M 25</b>	Grau de independência de componentes externos	X = 1 Médio

Diferentemente da 1.0, como na versão 2.0 há interação com a aplicação para o uso do mapa, o usuário final tem a sensação de que o sistema “demora” mais. Essa sensação é associada a exigência da aplicação de que haja um contato direto aplicação-usuário e, com isso, há a falsa sensação de aumento no tempo do provimento do serviço.

Apesar disso, o resultado obtido com a coleta das medidas permanece o mesmo.

## 2. Qual o grau de integridade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M19</b>	Grau de Integridade dos Dados	X = 2
	Transmitidos em Rede	Baixo
<b>M20</b>	Grau de Integridade dos Dados	X = 2
	Armazenados	Baixo

Assim como na versão 1.0, a versão 2.0 não implementa nenhum mecanismo para garantia da integridade dos dados transmitidos pelas diferentes partes envolvidas na entrega do serviço. Também não há garantias de integridade para os dados armazenados no dispositivo. Como a localização dos compromissos é adicionada a aplicação pelo próprio usuário, no próprio dispositivo, preocupações com os mecanismos de integridade foram postergadas para futuras versões. Além disso, quando existe uma transmissão de dados em uma rede TCP-IP, mecanismos de integridade já são implementados em camadas inferiores.

## 3. Qual o grau de manutenibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M16</b>	Grau de Documentação	X = 1
		Médio
<b>M17</b>	Grau de Esforço para Realizar	X = 1
	Manutenção do Sistema	Médio
<b>M18</b>	Grau de Esforço para Realizar	X = 1
	Manutenção ou Evolução do Código	Médio

Com relação ao grau de manutenibilidade, com a coleta das medidas é possível perceber que, mesmo com a existência de documentação, ainda há esforço extra para realizar a manutenção e

evolução do sistema. Indicando assim a necessidade de uma melhor documentação, de acordo com os desenvolvedores que participaram da coleta destas medidas.

4. Qual o grau de confiabilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M03</b>	Grau de Precisão	X = 0,72 Ou 72%
<b>M04</b>	Grau de Recall	X = 0,91 Ou 91%
<b>M07</b>	Grau de Acurácia	X = 0,66 Ou 66%
<b>M10</b>	Corretude na Execução da Adaptação	X = 1 Médio
<b>M05</b>	Grau de Especificidade	X = 87%
<b>M06</b>	Grau de Desempenho	X = 81%
<b>M09</b>	Grau de Adaptação	X = 92%
<b>M14</b>	Grau de Concordância	X = 1
<b>M15</b>	Grau de Independência ou Autonomia	X = M Médio
<b>M22</b>	Grau de Funcionamento em Background	X = M Médio

Com relação ao grau de confiabilidade, diferentemente da versão 1.0, a versão 2.0 obteve resultados considerados bons. Ainda há a geração de falsos positivos, porém, o problema identificado e que deve ser tratado em versões futuras consiste apenas em um atraso no provimento do serviço. O atraso ocorre em relação ao tempo necessário ao provimento do serviço e distância do usuário em relação ao local do compromisso (distâncias de até 50 metros para o local do compromisso e horários de até 4 minutos depois do horário de início).

Além disso, é importante destacar que agora a aplicação tem menos problemas relacionados à sincronização aplicação-agenda quando em *background*.

5. Qual o grau de proteção do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M11</b>	Grau de Compreensão no Uso dos Dados	X = 0 Alto
<b>M12</b>	Grau de Permissão no Uso dos Dados	X = 0 Alto
<b>M15</b>	Presença de Políticas de Segurança	X = 1

Por fim, o grau de proteção do sistema é considerado alto pois, diferentemente da versão anterior, por ter a possibilidade de alterar quais informações contextuais podem ser usadas pela aplicação e o momento de uso, torna a aplicação mais atraente para o usuário final.

### 5.2.3 GREat Mute v 3.0

Mesmo diante da evolução realizada entre as versões 1.0 e 2.0, e das melhorias obtidas e constatadas com a aplicação das medidas, é possível identificar a necessidade de uma nova evolução. Essa necessidade pode ser fundamentada em dois fatos: os resultados obtidos ainda não são os melhores possíveis, dado os valores de interpretação para cada medida coletada e, por fazer com que o usuário indique de maneira explícita a localização do seu compromisso em um mapa, o sistema perde em ubiquidade.

Com isso, a versão 3.0 continua fazendo uso do mesmo conjunto de informações contextuais, porém, a localização do compromisso é obtida pela aplicação de maneira automática, retirando esse dado da Google agenda.

Para a coleta das medidas na nova versão foi necessário realizar novos testes. Ao todo, 150 novos casos de teste foram feitos com dois usuários, utilizando os mesmos dispositivos dos testes das versões anteriores (*Samsung Galaxy S3* e *Samsung Galaxy S4*).

Nesta seção são apresentados os resultados obtidos pela coleta de dados na aplicação GREat Mute 3.0. Cada questão GQM, apresentada no Capítulo 4, é comentada de acordo com os resultados das medidas obtidas.

1. Qual o grau de disponibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
-----------	-------------	------------------

<b>M01</b>	Grau de Disponibilidade	X = 1
<b>M02</b>	Taxa de Erros	X = 0
<b>M08</b>	Tempo de Adaptação	X = T = 63 milissegundos Dp = 112 milissegundos
<b>M 11</b>	Discrição dos mecanismos de segurança	X = 3 Baixo
<b>M 23</b>	Grau de Independência da Bateria	X = 1 Médio
<b>M 24</b>	Grau de influência sobre a bateria	X = 1 Médio
<b>M 25</b>	Grau de independência de componentes externos	X = 1 Médio

Os resultados obtidos para o grau de disponibilidade são semelhantes aos resultados das versões 1.0 e 2.0. A taxa de erros continua nula e o sistema está disponível sempre que requisitado. Porém, diferentemente da versão 2.0, não há mais a necessidade de interação direta do usuário com o mapa da aplicação. Isso fez com que a sensação de disponibilidade do sistema, por parte do usuário final, parecesse mais imediata. Implicando em uma melhoria não captada pela coleta das medidas.

## 2. Qual o grau de integridade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M19</b>	Grau de Integridade dos Dados Transmitidos em Rede	X = 1 Médio
<b>M20</b>	Grau de Integridade dos Dados Armazenados	X = 1 Médio

Como mencionado anteriormente, apesar de não existir a necessidade da implementação de mecanismos de integridade em serviços web, devido a existência de mecanismos de integridade da pilha TCP-IP, houveram diferenças na coleta das medidas de grau de integridade. Essas diferenças são proporcionadas pela nova implementação dos recursos de configuração da aplicação, providos ao usuário. Onde, o usuário do sistema é devidamente informado e notificado de que a aplicação necessita estar em comunicação direta com a google agenda.

Assim, com o uso de permissões *Android*, é possível fazer com que a aplicação faça sincronização de contas automaticamente, de tempos em tempos, garantindo que os dados conhecidos pela aplicação, dentro do dispositivo, são condizentes com os dados que estão na agenda do google, associada a conta de email do usuário. Logo, do ponto de vista do usuário final, há melhorias nesse aspecto.

### 3. Qual o grau de manutenibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M16</b>	Grau de Documentação	X = 2 Alto
<b>M17</b>	Grau de Esforço para Realizar Manutenção do Sistema	X = 0 Baixo
<b>M18</b>	Grau de Esforço para Realizar Manutenção ou Evolução do Código	X = 0 Baixo

Assim como na versão anterior (2.0) com coleta das medidas é possível perceber que, mesmo com a realização de melhorias na documentação, foi possível reduzir o esforço para realizar a manutenção e evolução do sistema.

### 4. Qual o grau de confiabilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M03</b>	Grau de Precisão	X = 0,68 Ou 68%
<b>M04</b>	Grau de Recall	X = 0,91 Ou 91%
<b>M07</b>	Grau de Acurácia	X = 0,71 Ou 71%
<b>M10</b>	Corretude na Execução da Adaptação	X = 2 Alto
<b>M05</b>	Grau de Especificidade	X = 87%
<b>M06</b>	Grau de Desempenho	X = 91%
<b>M09</b>	Grau de Adaptação	X = 92%

<b>M14</b>	Grau de Concordância	X = 1
<b>M15</b>	Grau de Independência ou Autonomia	X = M Médio
<b>M22</b>	Grau de Funcionamento em Background	X = 2 Alto

Com relação ao grau de confiabilidade, não houveram mudanças significativas em relação a geração de falsos positivos ou falsos negativos. As melhorias estão em: funcionar de maneira independente ou autônoma, pois não necessitar mais da interação explícita usuário-mapa e por funcionar perfeitamente em *background* (e.g., quando o dispositivo se encontra em modo de descanso ou bloqueado).

5. Qual o grau de proteção do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M11</b>	Grau de Compreensão no Uso dos Dados	X = 0 Alto
<b>M12</b>	Grau de Permissão no Uso dos Dados	X = 0 Alto
<b>M15</b>	Presença de Políticas de Segurança	X = 1

Por fim, para o grau de proteção do sistema, não houveram alterações em relação a coleta da versão anterior (2.0)

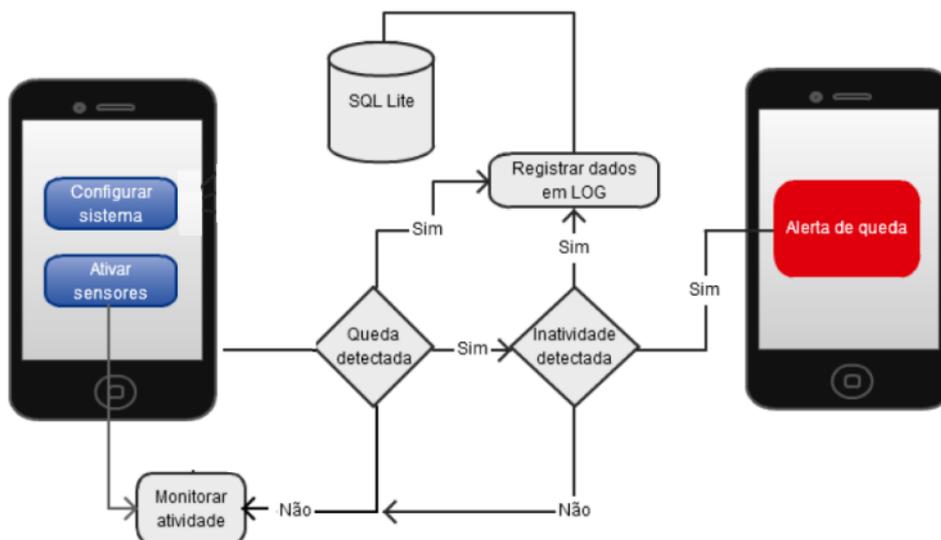
### 5.3 fAlert

O *fAlert* (*Fall Alert*) é um protótipo de aplicação de *software*, para *smartphones* com sistema operacional *android*. A aplicação atua monitorando de forma inteligente pessoas que necessitam de cuidados especiais, como idosos e deficientes físicos (PIVA et al 2014).

O sistema consiste em uma aplicação ubíqua que, em tempo real, utiliza os dados fornecidos pelos sensores presentes nos *Smartphones* para detectar possíveis quedas, tais como

acelerômetro, microfone e magnetômetro. A aplicação permite a configuração de telefones, e-mails e endereços de emergência para que, uma vez detectado um padrão de queda, um alerta seja enviado para uma entidade pré-configurada.

A Figura 21 mostra o esquema de representação do sistema de detecção de quedas, mostrando as configurações do dispositivo e o processo de ativação do monitoramento dos dados dos sensores.



**Figura 21: Representação do funcionamento do sistema fAlert (PIVA et al 2014)**

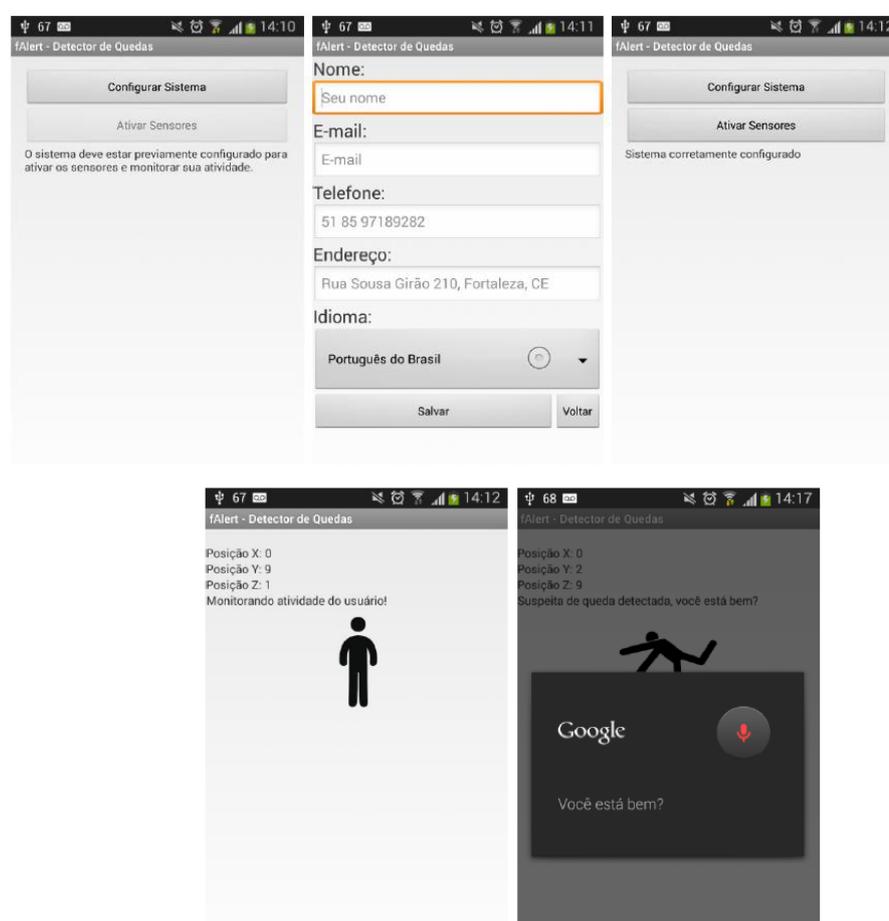
O protótipo de aplicação *fAlert* foi desenvolvido com base no dispositivo *Samsung Galaxy S3* que possui a seguinte configuração, apresentada na Tabela 12.

**Tabela 12: Configuração técnica do aparelho utilizado [PIVA et al 2014]**

Dimensões	136.5 x 70.6 x 8.6 mm
Tamanho da Tela	4.8 polegadas de 720 x 1280 pixels
Peso	133g
Tecnologias sem Fio	Wifi 802.11 b/g/n, 2.5G, 3G, Bluetooth, NFC e GPS
Armazenamento	16GB
Memória RAM	1GB
Processador	Quad-core 1.4 GHz Cortex-A9
Bateria	Li-Ion 2100 mAh
Qualidade do Som	-90.3 dB e -92.6 dB

Sensores	Acelerômetro, microfone, giroscópio, proximidade, magnetômetro, barômetro
Sistema Operacional	Android versão 4.3 Jelly Bean

A aplicação possui uma interface gráfica intuitiva, apresentada na Figura 22, possuindo apenas duas funcionalidades principais, que facilitam a compreensão e o uso do sistema. A primeira funcionalidade é utilizada para as configurações de contato e idioma, permitindo a seleção de três idiomas: Português do Brasil, Espanhol e Inglês. A segunda funcionalidade é utilizada para a ativação do monitoramento das atividades do usuário por meio dos sensores do dispositivo, que somente estará disponível após a configuração dos dados de contato.



**Figura 22: Interface gráfica do fAlert**

A interface para o monitoramento exibe informações dos três eixos (x,y e z) do acelerômetro, permitindo, assim, mostrar para o usuário que os dados estão sendo obtidos corretamente a medida que ele se move.

### 5.3.1 Mecanismo para a Detecção de Quedas

As soluções baseadas em *smartphones* utilizam dois possíveis mecanismos para a detecção de quedas: reconhecimento de padrões ou *trheshold* (FUDICKAR et al 2014). No primeiro caso, o sistema se comunica com sensores externos ao dispositivo, que podem auxiliar o reconhecimento de padrões que identifiquem as atividades do usuário. Assim, é possível fazer com que o sistema aprenda sobre o comportamento do usuário e seja capaz de diferenciar entre situações cotidianas e de perigo. No segundo caso, o sistema utiliza apenas os sensores presentes no *smartphone* para o processo de detecção. Com isso, é possível baratear o custo e prover o serviço sem a necessidade de outro hardware. Sendo este segundo o mecanismo adotado pelo *fAlert*.

O *fAlert* utiliza os dados obtidos por meio do vetor de aceleração linear, produzido pelo acelerômetro presente no dispositivo móvel e identificado pela Equação 1 apresentada a seguir:

$$|A_T| = \sqrt{|A_x|^2 + |A_y|^2 + |A_z|^2}$$

**Equação1: Fórmula para Cálculo da Aceleração Linear do Dispositivo**

Com base na fusão da informação dos três eixos (x,y e z) do acelerômetro e os dados obtidos pela leitura do magnetômetro, a aplicação pode identificar os casos onde o usuário se encontra em queda livre, por um curto período de tempo, seguido da ocorrência de impacto. Caracterizando assim, uma possível ocorrência de queda.

Para que a queda seja confirmada, verifica-se se ocorreu uma variação brusca da aceleração do dispositivo bem como a posição na qual o dispositivo se encontra. Caso o dispositivo esteja em um ângulo entre 0 e 45 graus em relação ao chão, a possibilidade de queda é identificada.

Depois de identificada uma possível queda, o sistema ativa o *Google Speech Recognizer* para verificar se o usuário se encontra bem ou não. Isso é feito por meio do lançamento de uma janela de diálogo, onde o usuário precisa interagir com o dispositivo utilizando a voz em um tempo mínimo de 25 segundos. A interação deve ocorrer em voz alta, por meio do uso de palavras-chave.

No caso do usuário estar bem, a aplicação volta ao seu estado inicial de monitoramento das atividades do usuário. Caso contrário, quando não há reconhecimento de voz ou o usuário fala que não está bem, o sistema gera o alerta, caracterizado pela realização de uma ligação telefônica para um contato previamente configurado, na opção de menu “*Configurar Sistema*”.

De acordo com os sensores presentes no dispositivo do usuário, há variação nas informações disponíveis para a realização dos cálculos feitos pelo sistema e, conseqüentemente

na precisão das detecções realizadas. Os dados disponíveis de acordo com cada sensor são apresentados na Tabela 13.

**Tabela 13: Informações providas pelos Sensores Utilizados para Detecção**

Informações Utilizadas para a Detecção	Acelerômetro	Acelerômetro e Magnetômetro	Acelerômetro, Magnetômetro e Microfone
Queda Livre	X	X	X
Impacto	X	X	X
Movimento Pós Impacto	X	X	X
Estabilidade	X	X	X
Orientação do Dispositivo		X	X
Confirmação Sonora			X

### 5.3.2 Testes Realizados

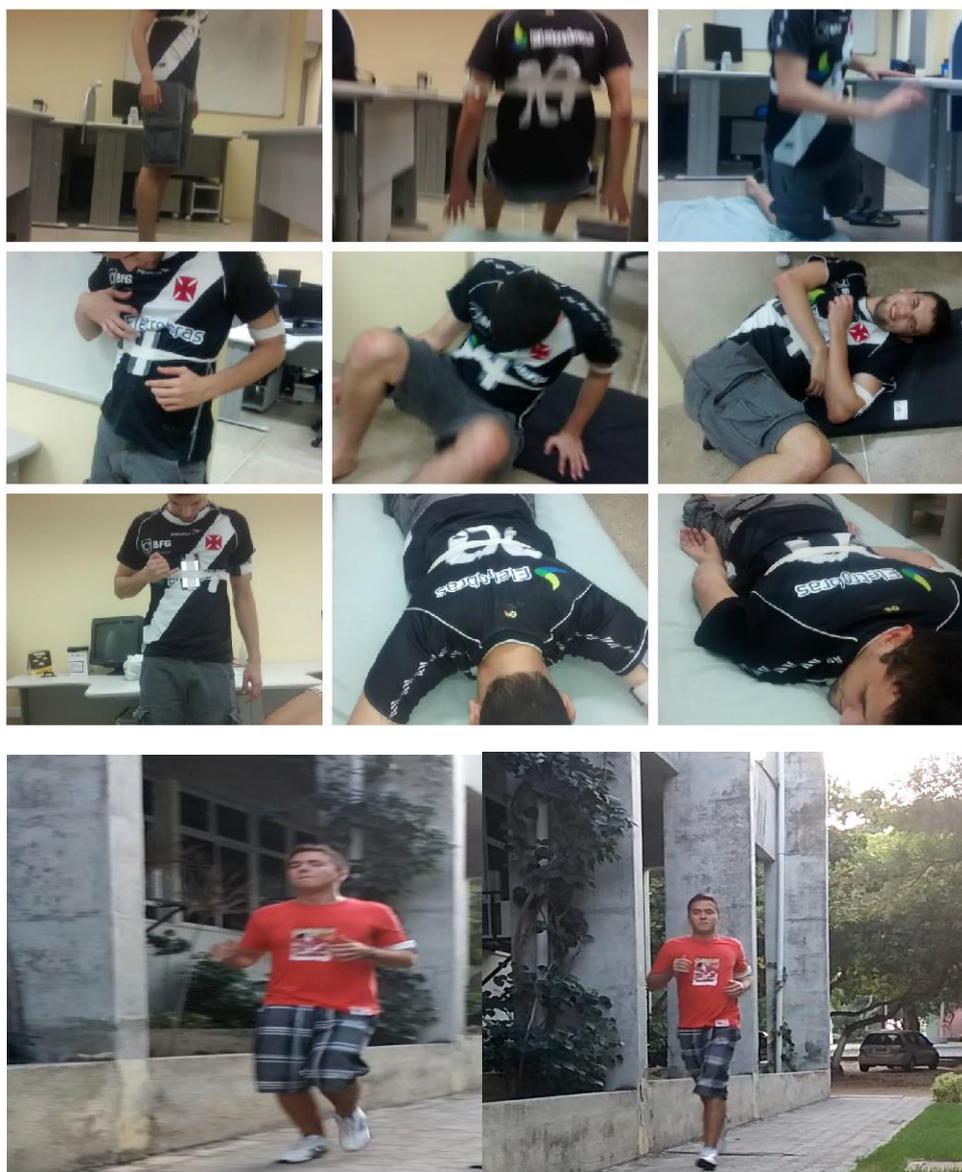
Os testes realizados para a detecção de quedas levam em consideração diversas atividades cotidianas do usuário que podem ser detectadas como possíveis quedas, são exemplos: andar, correr, sentar, levantar e deitar.

Para produzir resultados mais precisos foram realizados 1.140 testes em quatro usuários jovens e saudáveis, utilizando os dispositivos *Samsung Galaxy S3 I9300*, *Samsung Galaxy S3 Slim Duos* e *LG G2*, colocados no peito, braço e bolso, respectivamente, ilustrados na Figura 23. Nos testes específicos para os casos de queda foi utilizado um colchão de espuma localizado no chão.



**Figura 23: Dispositivos espalhados pelo corpo do usuário**

É importante salientar também que as quedas foram divididas em categorias: quedas frontais, laterais e para trás. Os testes foram realizados em um ambiente fechado bem como em um ambiente aberto, como ilustram as imagens na Figura 24.



**Figura 24: Casos de queda**

A quantidade de testes realizados por pessoa é apresentada de maneira categorizada na Tabela 14, a seguir:

**Tabela 14: Quantidade de testes por categoria**

Atividades	<i>Indoor</i>	<i>Outdoor</i>	Frontal	Lateral	Para trás
Andar	15	15	-	-	-
Correr	0	30	-	-	-
Pular	30	0	-	-	-
Subir e Descer Escadas	30	0	-	-	-
Sentar Lento	30	0	-	-	-
Sentar Rápido	30	0	-	-	-
Deitar Lento	30	0	10	10	10
Deitar Rápido	30	0	10	10	10
Cair Lento	30	0	10	10	10
Cair Rápido	30	0	10	10	10

Como pode-se observar na Tabela 14, para cada atividade, 30 testes foram realizados. Esse número foi considerado o suficiente, pois com essa mesma quantidade foi possível definir o *threshold* para o algoritmo de detecção. Para as ações de deitar e cair os testes foram subdivididos de acordo com a direção do movimento.

Diferentemente do GREat Mute, onde foram coletadas todas as medidas, para a avaliação da confiança no funcionamento do sistema *fAlert* é utilizado um subconjunto das medidas de qualidade apresentadas no Capítulo 4. Esse subconjunto foi escolhido com base em diversos trabalhos (e.g.,) encontrados na literatura que indicam quais aspectos são importantes e devem ser levados em consideração ao se avaliar esse tipo de sistema. Adaptadas para a avaliação do *fAlert*, as principais medidas utilizadas são apresentadas em detalhes na Tabela 15. A coleta de todas as medidas foi realizada apenas na última versão do sistema (versão com os sensores de acelerômetro, magnetômetro e microfone) e os resultados são apresentados no fim desta seção.

**Tabela 15: Medidas utilizadas para avaliar o fAlert**

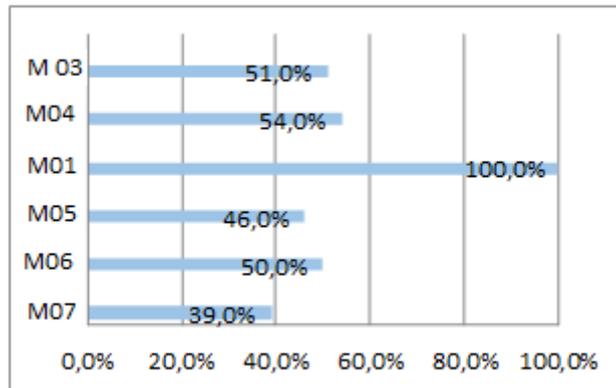
<i>Id</i>	<i>Nome</i>	<i>Descrição</i>	<i>Função de Medição</i>	<i>Interpretação</i>
M03	Grau de Precisão	Representa a capacidade do sistema de detectar somente quedas, avaliando, para tal, a fração de contextos sentidos ou inferidos que são relevantes para o sistema.	$\frac{P}{P + FP}$	Quanto mais próximo de 1, melhor.
M04	Grau de Sensibilidade (Recall)	Representa a capacidade do sistema de detectar quedas, avaliando, para tal, a fração de contextos relevantes para o sistema que foram corretamente sentidos ou inferidos.	$\frac{P}{P + FN}$	Quanto mais próximo de 1, melhor.

M 01	Grau de Disponibilidade	Representa a capacidade do sistema de prestar serviço quando requisitado.	$\frac{TmF}{TmF + TmR}$	Quanto mais próximo de 1, melhor.
M 05	Grau de Especificidade	Representa a fração entre os alertas não gerados, decorrentes da ausência de quedas e a soma destes com o total de alertas gerados para quedas que não ocorreram, complementando assim, informações obtidas por meio da coleta do <i>Grau de Precisão</i> .	$\frac{N}{N + FP}$	Quanto mais próximo de 1, melhor.
M 06	Grau de Desempenho	Representa a média ponderada entre o <i>Grau de Precisão</i> e o <i>Grau de Sensibilidade (Recall)</i> .	$\frac{2 * (\frac{P}{P + FP} * \frac{P}{P + FN})}{\frac{P}{P + FP} + \frac{P}{P + FN}}$	Quanto mais próximo de 1, melhor.
M 07	Grau de Acurácia	Representa a fração de quedas corretamente sentidos ou inferidos	$\frac{P + N}{FP + P + FN + N}$	Quanto mais próximo de 1, melhor.

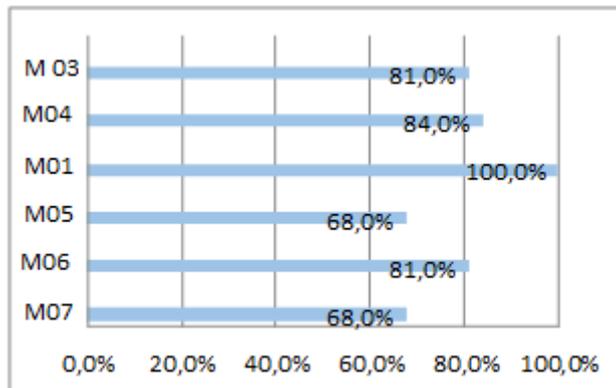
TmF e TmR representam respectivamente o tempo médio entre falhas e o tempo médio para reparar as falhas. P representa a ocorrência de uma queda detectada corretamente pelo sistema. Por outro lado, N representa que não houve queda e o sistema não detectou. Já FP representa a detecção de uma queda que não ocorreu e FN representa que houve queda, mas o sistema não identificou. Esses critérios, quando combinados, podem fornecer informações importantes sobre a acurácia do sistema, precisão, sensibilidade e outros.

Alguns dos resultados obtidos estão ilustrados nas Figuras 25, 26 e 27. Os gráficos apresentados nessas figuras representam medidas que impactam no grau de confiança do sistema *fAlert*. Isso é possível por meio da utilização de um sensor ou da combinação de dois ou mais sensores, configurados da seguinte forma: (i) somente acelerômetro; (ii) acelerômetro e microfone; e (iii) acelerômetro, microfone e magnetômetro.

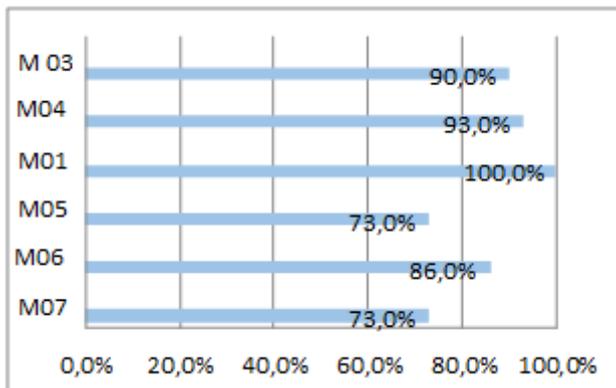
A avaliação da confiança do sistema levou em consideração essas três configurações, pois o sistema possibilita ao usuário, durante a configuração inicial, escolher quais sensores o sistema pode utilizar.



**Figura 25: (i) Resultados obtidos com o uso do acelerômetro para dispositivo atado ao peito**



**Figura 26: (ii) Resultados obtidos com o uso do acelerômetro e microfone para dispositivo atado ao peito**



**Figura 27: (iii) Resultados obtidos com o uso do acelerômetro, magnetômetro e microfone para dispositivo atado ao peito**

Como pode-se observar nas Figuras 25, 26 e 27, os valores obtidos para a medida M03 são inferiores aos da medida M04 para as três configurações utilizadas, enquanto a medida M01 permanece constante.

Para as medidas M05, M06 e M07, os valores alcançados para cada nova configuração foram sempre maiores que os obtidos com a configuração anterior. Destacando-se as medidas M06 e M07 onde o aumento alcançado entre a coleta da primeira e segunda configuração foi significativamente maior que o aumento atingido com a mudança da segunda para terceira. A análise aprofundada de todos os resultados é apresentada em detalhes na Seção 5.4.3, a seguir.

### 5.3.3 Interpretação dos Resultados

Nesta seção são apresentados os resultados obtidos com a coleta das medidas na aplicação *fAlert*. Com a análise dos resultados alcançados é possível interpretar as questões e o objetivo de medição definidos nesta pesquisa.

O valor obtido para cada uma das medidas coletadas nas três configurações utilizadas, e para as três diferentes posições nas quais os dispositivos se encontravam no corpo do usuário, é apresentado graficamente nas Figuras 28-33.

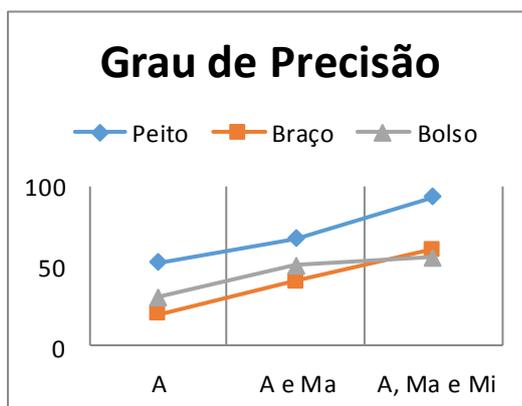


Figura 28: Resultados obtidos para a medida precisão

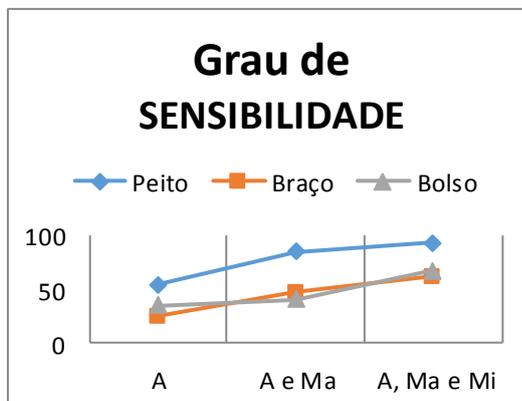


Figura 29: Resultados obtidos para a medida sensibilidade

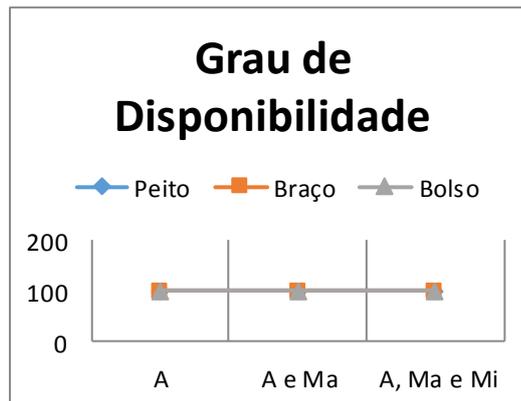


Figura 30: Resultados obtidos para a medida disponibilidade

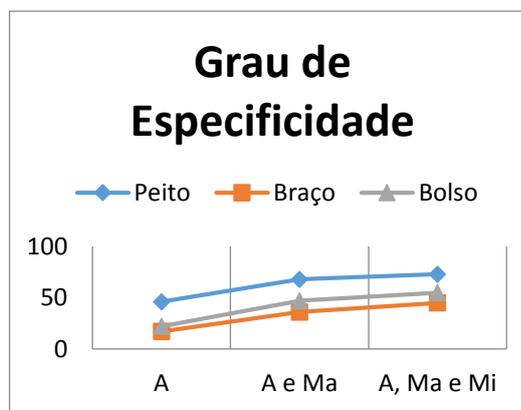


Figura 31: Resultados obtidos para a medida especificidade

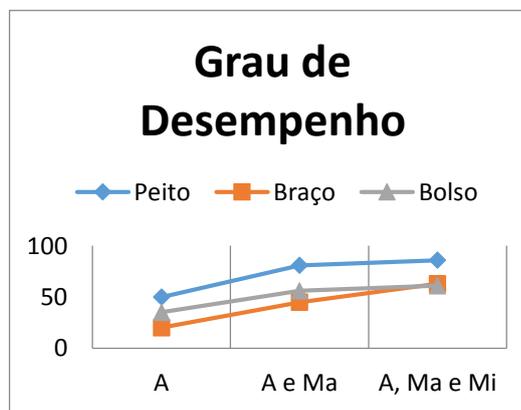
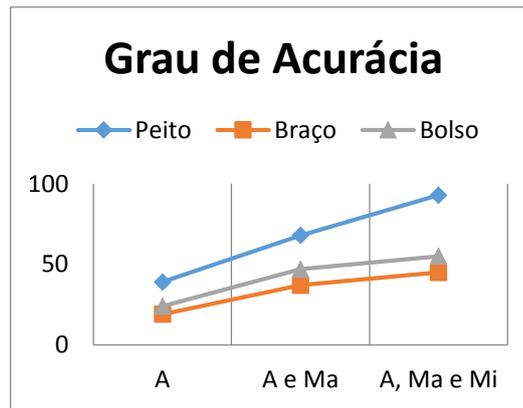


Figura 32: Resultados obtidos para a medida grau de desempenho



**Figura 33: Resultados obtidos para a medida acurácia**

Para as três diferentes posições de alocação do dispositivo (braço, peito e bolso), com a coleta da medida M03. Na primeira configuração (apenas o sensor acelerômetro), é possível perceber que o número de falsos positivos (FP) gerados pelo sistema é significativo. Essa identificação de FP é facilmente percebida porque a função de medição de M03 possibilita ao avaliador identificar a ocorrência de alertas para situações que não são quedas (e.g. sentar, levantar, deitar).

Além da quantidade significativa de falsos positivos gerados, esta primeira configuração gerou também um número considerável de falsos negativos. Nesse caso, independentemente da posição do dispositivo, no pior caso, 60% das quedas não foram detectadas, contribuindo negativamente para a medida de *Sensibilidade*. Os resultados obtidos reforçam a necessidade do uso da combinação de outros sensores do dispositivo.

Na segunda configuração, utilizando os sensores de acelerômetro e microfone, os valores coletados para as medidas foram melhores. O número de falsos positivos reduziu em aproximadamente 30% para a maioria das medidas. Porém, não alcançaram os mesmos níveis dos sistemas de detecção de quedas encontrados nos trabalhos relacionados. Além disso, falsos positivos foram gerados para as ações de sentar, levantar e deitar.

Ainda assim, os resultados obtidos são animadores pois, nos demais trabalhos encontrados, para que esse nível de precisão fosse atingido eram necessárias, muitas vezes, informações obtidas por sensores externos.

Finalmente, na terceira configuração (acelerômetro, microfone e magnetômetro) foi possível eliminar as ações de sentar e levantar, pois não houve a geração de falsos positivos. Para as ações de deitar e cair, os valores coletados com as medidas obtiveram melhores valores do que as duas primeiras configurações, aumentando a precisão e a sensibilidade do sistema.

Por fim, com a distribuição dos dispositivos pelo corpo do usuário, foi possível identificar que o peito é a melhor posição para a alocação do *smartphone* para a detecção de quedas. Por

manter um padrão de movimentos mais estável, independentemente do tipo de queda, o algoritmo de detecção funciona melhor.

Situações de queda livre seguidas de impacto são bem identificadas pelo dispositivo alocado no braço, porém, a orientação do dispositivo após o impacto e o ângulo formado entre o celular e o chão influenciam o algoritmo erroneamente.

Quando alocado no bolso, o sistema também gera uma quantidade significativa de falsos positivos. Quando o usuário apenas senta bruscamente o sistema identifica a queda livre e o impacto, após isso, dependendo da altura do usuário, o celular se aloca no bolso de maneira horizontal, caracterizando situações onde a orientação do dispositivo corresponde a uma queda. Além disso, quando a detecção ocorre de maneira correta, a distância da boca até o bolso impede a confirmação sonora, causando atrasos ou erros no processo de detecção.

Seguindo o mesmo método de apresentação dos resultados obtidos apresentados na seção anterior (*GREat Mute*), para a versão do *fAlert* com o uso dos três sensores e dispositivo alocado ao peito (melhor configuração e posição de alocação), as respostas para as questões GQM são:

1. Qual o grau de disponibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M 01</b>	Grau de Disponibilidade	X = 1
<b>M 02</b>	Taxa de Erros	X = 0
<b>M 08</b>	Tempo de Adaptação	X = T = 3 segundos
<b>M 11</b>	Discrição dos mecanismos de segurança	X = 3 Baixo
<b>M 23</b>	Grau de Independência da Bateria	X = 2 Alto
<b>M 24</b>	Grau de influência sobre a bateria	X = 1 Médio
<b>M 25</b>	Grau de independência de componentes externos	X = 2 Alto

O grau de disponibilidade do *fAlert* em sua configuração com o uso de três sensores pode ser considerado alto. O sistema não indica a existência de erros e está disponível sempre que requisitado. Além disso, o sistema não é afetado por índices baixos de bateria. E, de acordo com os usuários, o tempo de adaptação, mesmo sendo em segundos, é tolerável dado a sensibilidade

das circunstâncias de contexto (e.g., 3 segundos é um tempo aceitável para um sistema que consegue distinguir ações de deitar e cair).

2. Qual o grau de integridade do sistema?

Id	Nome	Resultado
M19	Grau de Integridade dos Dados Transmítidos em Rede	N/A
M20	Grau de Integridade dos Dados Armazenados	N/A

Com relação ao grau de integridade, o *fAlert* não armazena dados relativos ao processo de detecção. As verificações são feitas de maneira contínua, sem aprendizado ou armazenamento de dados. Logo, não há aplicabilidade para a medida M20. O mesmo vale para a medida M19.

3. Qual o grau de manutenibilidade do sistema?

Id	Nome	Resultado
M16	Grau de Documentação	X = 0 Baixo
M17	Grau de Esforço para Realizar Manutenção do Sistema	X = 1 Médio
M18	Grau de Esforço para Realizar Manutenção ou Evolução do Código	X = 1 Médio

Com relação ao grau de manutenibilidade, com a coleta das medidas é possível perceber que, mesmo com a inexistência de documentação, o esforço extra para realizar a manutenção e evolução do sistema é menor que o esperado. A explicação associada a isto está na simplicidade do código. Sendo este resultado uma exceção ao esperado.

4. Qual o grau de confiabilidade do sistema?

Id	Nome	Resultado
M 03	Grau de Precisão	X = 0,90 Ou 90%

<b>M 04</b>	Grau de Recall	X = 0,93 Ou 93%
<b>M 07</b>	Grau de Acurácia	X = 0,73 Ou 73%
<b>M10</b>	Corretude na Execução da Adaptação	X = 2 Alto
<b>M 05</b>	Grau de Especificidade	X = 73%
<b>M 06</b>	Grau de Desempenho	X = 86%
<b>M 09</b>	Grau de Adaptação	X = 89%
<b>M 14</b>	Grau de Concordância	X = 1
<b>M 15</b>	Grau de Independência ou Autonomia	X = M Médio
<b>M22</b>	Grau de Funcionamento em Background	X = 0 Baixo

Segundo os usuários que realizaram os testes e as pessoas envolvidas no desenvolvimento do sistema, as medidas relativas à confiabilidade estão entre as mais importantes para a avaliação da confiança no funcionamento. Nesse aspecto, o *fAlert* obteve resultados satisfatórios pois, não exigindo custo extra com aquisição de novos equipamentos de hardware ou cursos para instalação e aprendizado do uso de software, é possível obter um sistema com taxas de acerto equivalentes a soluções, encontradas no mercado, como mais “robustas”.

Além disso, é importante destacar que, com a coleta das medidas é possível quantificar o ganho obtido com cada novo sensor associado ao processo de detecção.

#### 5. Qual o grau de proteção do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M11</b>	Grau de Compreensão no Uso dos Dados	X = 0 Alto
<b>M12</b>	Grau de Permissão no Uso dos Dados	X = 0 Alto
<b>M15</b>	Presença de Políticas de Segurança	X = 0 Nenhuma

Por fim, o grau de proteção possui resultados positivos. Apesar de não existir a implementação de nenhuma política de segurança, a aplicação é altamente satisfatória em relação ao uso dos dados do usuário e a permissão para a utilização destes.

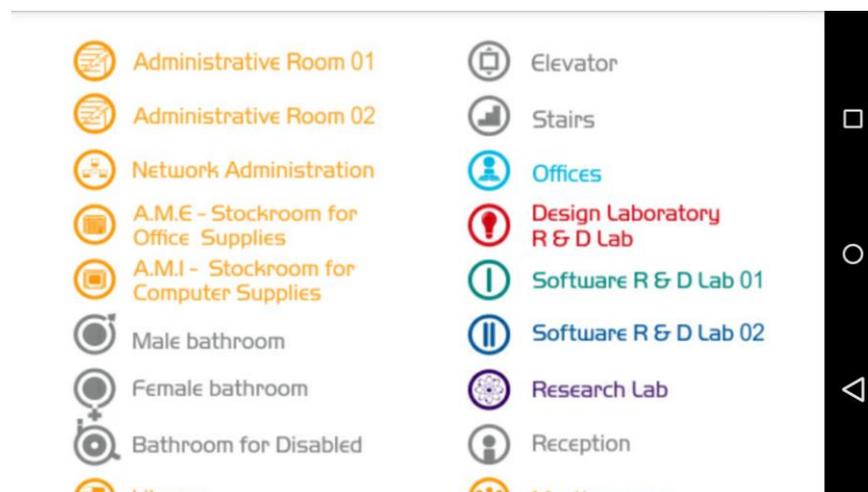
Assim, os resultados obtidos com a coleta de todo o conjunto de medidas reforçam os resultados obtidos com as coletas parciais, indicando bons resultados para a configuração com o uso dos três sensores, quando comparado aos resultados obtidos em sistemas semelhantes, encontrados em trabalhos relacionados.

Mesmo diante desse cenário, é possível identificar oportunidades de evolução e, conseqüentemente, trabalhos futuros. Essa afirmação tem por base o fato de que as medidas apresentadas neste trabalho foram definidas e avaliadas tendo como foco apenas o domínio dos sistemas para dispositivos móveis, não havendo indícios de que possam ser aplicadas em outros domínios. Acarretando, possivelmente, na necessidade de novas avaliações.

Além disso, como mencionado em (FERREIRA ET AL 2015), o *fAlert* é uma prova de conceito (PoC, do inglês *proof of concept*). A PoC serve como modelo prático para confirmar os conceitos teóricos estabelecidos neste trabalho, porém não é possível fazer afirmações sobre a aplicação das medidas em sistemas reais.

## 5.4 GREAt Tour

O GREAt Tour é uma aplicação *Android* que serve como um guia de visitas do laboratório GREAt. A aplicação, sensível ao contexto, captura a posição do visitante em tempo real e exibe informações do local onde ele se encontra, bem como realiza adaptações com base nessa localização.

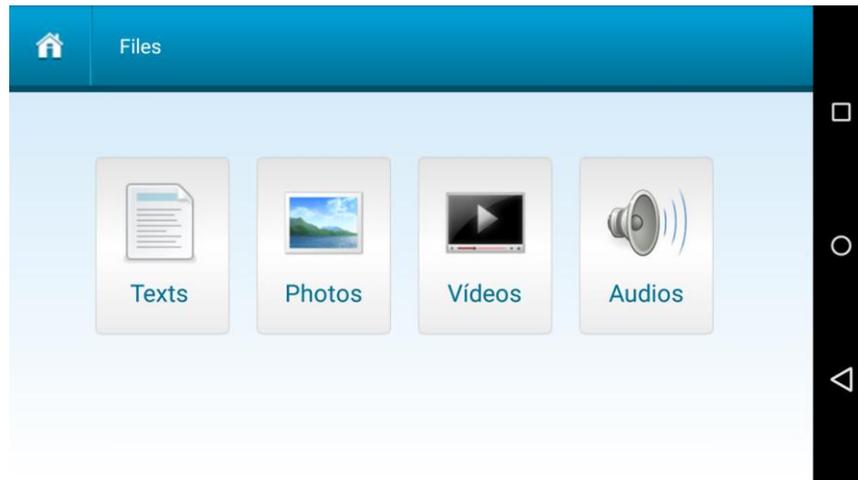


**Figura 34: Telas do GREAt Tour**

A aplicação encontra-se atualmente em estado de evolução para a versão 3.0. Nesta dissertação, a versão avaliada é a 2.0, que pode ser encontrada para download no seguinte link: <http://www.great.ufc.br/ctqs/index.php/producao/aplicacoes/78-artigosctqs/102-greattour>.

Em sua versão 2.0, a aplicação deve permitir que o usuário faça uma visita às alocações do GREAt (e.g., salas, laboratórios) de forma guiada, fornecendo informações a respeito dos ambientes a medida que o usuário se move.

Dentre as informações fornecidas pela aplicação encontram-se: vídeos, imagens, áudio, artigos, papers, dissertações, tese e outros, respeitando sempre o critério de exibi-las de acordo com o ambiente no qual o usuário se encontra.



Dentre os requisitos funcionais e não-funcionais, encontrados no *template* de especificação de requisitos da aplicação, encontram-se:

- 1 – O sistema deve ser capaz de guiar um visitante pelas alocações do GREat;
- 2 – O sistema deve ser capaz de fazer leituras de QR codes, bem como de tag's NFC, pois as informações de localidade são associadas pelos códigos presentes nas supracitadas tecnologias;
- 3 – O sistema deve fazer requisições ao banco de dados via *web service*, de forma rápida, para garantir uma experiência não cansativa da aplicação;
- 4 – O sistema deverá retornar informações referentes ao local onde o visitante se encontra;
- 5 – A aplicação deverá fornecer ao usuário um mapa do local atual, indicando em que ponto ele se encontra ao ler a localização;
- 6 – O sistema deve ser de fácil utilização;
- 7 – O sistema deve ser desenvolvido para Android OS;
- 8 – Os pacotes e classes da aplicação devem ser organizados de tal forma a minimizar o tempo de manutenção, caso necessário;
- 9 – Um novo usuário deve ser capaz de utilizar a aplicação desde que seja feito um cadastro prévio.

É possível perceber, observando requisitos como o 1, 4, 5 e 8 que, as medidas definidas no Capítulo 4, possuem relação direta com a garantia destes.

Diante desse cenário, as medidas definidas no capítulo anterior, são coletadas na versão 2.0 desta aplicação. Para tanto, foram realizados 50 casos de teste com os

dispositivos *Samsung Galaxy Dual Prime e Nexus 4*. Os resultados obtidos são apresentados a seguir como respostas para as questões GQM apresentadas no Capítulo anterior.

1. Qual o grau de disponibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M 01</b>	Grau de Disponibilidade	X = 1
<b>M 02</b>	Taxa de Erros	X = 0,3
<b>M 08</b>	Tempo de Adaptação	X = T = 1 segundo
<b>M 11</b>	Discrição dos mecanismos de segurança	X = 1 Médio
<b>M 23</b>	Grau de Independência da Bateria	X = 1 Alto
<b>M 24</b>	Grau de influência sobre a bateria	X = 3 Baixo
<b>M 25</b>	Grau de independência de componentes externos	X = 3 Baixo

O grau de disponibilidade do *GREat Tour* em sua versão 2.0 não é considerado o melhor possível, segundo as medidas coletadas. O sistema indica a existência de erros, por algumas vezes houve erros de login, geração do conteúdo e até mesmo inicialização do sistema. Além disso, o mecanismo de autenticação não é discreto o suficiente, pois, tentativas de “logar” na aplicação demoravam mais do que o aceitável, segundo o usuário final. Por fim, a alta dependência da aplicação com relação a componentes externos (e.g., qrcode), torna a disponibilidade um aspecto ainda mais crítico, a ser tratado em futuras versões.

2. Qual o grau de integridade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M19</b>	Grau de Integridade dos Dados Transmitidos em Rede	NA
<b>M20</b>	Grau de Integridade dos Dados Armazenados	NA

Assim como mencionado nos estudos de caso anteriores, os mecanismos de integridade são implementados em outras camadas do protocolo TCP-IP.

3. Qual o grau de manutenibilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M16</b>	Grau de Documentação	X = 0 Alto
<b>M17</b>	Grau de Esforço para Realizar Manutenção do Sistema	X = 2 Baixo
<b>M18</b>	Grau de Esforço para Realizar Manutenção ou Evolução do Código	X = 2 Baixo

Com relação ao grau de manutenibilidade, a aplicação obteve os melhores resultados dentre as três aplicações analisadas. A documentação da aplicação possui um nível de conteúdo muito satisfatório e, segundo os desenvolvedores envolvidos, o esforço para evoluir e manter a aplicação é adequado.

4. Qual o grau de confiabilidade do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M03</b>	Grau de Precisão	X = 0,94 Ou 94%
<b>M04</b>	Grau de Recall	X = 0,98 Ou 98%
<b>M07</b>	Grau de Acurácia	X = 0,92 Ou 92%
<b>M10</b>	Corretude na Execução da Adaptação	X = 2 Alto
<b>M05</b>	Grau de Especificidade	X = 89%
<b>M06</b>	Grau de Desempenho	X = 96%
<b>M09</b>	Grau de Adaptação	X = 97%
<b>M14</b>	Grau de Concordância	X = 1
<b>M15</b>	Grau de Independência ou Autonomia	Médio

<b>M22</b>	Grau de Funcionamento em Background	X = 0 Baixo
------------	--	----------------

Apesar dos resultados negativos, obtidos com a coleta das medidas de disponibilidade, as respostas obtidas para as questões relativas à confiabilidade, são animadoras. A aplicação obteve resultados próximos do melhor possível, para a maioria das medidas. Indicando que, mesmo não provendo serviço sempre que requisitado, a conformidade com o esperado, quando não ocorrem problemas de disponibilidade, proporcionam um ótimo auxílio para o aumento do grau de confiança no funcionamento do sistema por parte do usuário.

5. Qual o grau de proteção do sistema?

<b>Id</b>	<b>Nome</b>	<b>Resultado</b>
<b>M11</b>	Grau de Compreensão no Uso dos Dados	X = 0 Alto
<b>M12</b>	Grau de Permissão no Uso dos Dados	X = 0 Alto
<b>M15</b>	Presença de Políticas de Segurança	X = 1

Por fim, o grau de proteção possui resultados positivos. Além da implementação do mecanismo de login, a aplicação é altamente satisfatória em relação ao uso dos dados do usuário e a permissão para a utilização destes. É importante destacar aqui o uso não apenas das informações do usuário da aplicação, mas, também, dos ambientes e pessoas envolvidas nele. Aspecto este não constatado pelas medidas, mas sim pelas pessoas envolvidas no momento da coleta.

## 5.5 Considerações Finais

Com o intuito de avaliar a utilidade das 25 medidas propostas no Capítulo 4, este capítulo apresentou os resultados da coleta delas em três diferentes aplicações. Com os resultados obtidos foi possível identificar pontos de melhoria nos três sistemas. Tais pontos foram apresentados e discutidos nas seções 5.2, 5.3 e 5.4 e, possibilitaram o desenvolvimento de novas

e melhores versões dos sistemas avaliados, no que tange a confiança do usuário em seu funcionamento.

Evidenciando, com isso, indícios de que as medidas propostas nesta dissertação são, de fato, capazes de avaliar a característica de qualidade *confiança no funcionamento*.

Além disso, buscando contribuir para a avaliação das medidas propostas, alguns critérios foram levados em consideração. Esses critérios podem ser vistos na Tabela 16 adaptada de (KOSCIANSKI e SOARES 2007) e, segundo o autor, servem de critérios qualitativos para avaliar medidas de qualidade.

**Tabela 16: Critérios qualitativos para avaliar medidas**

<b>Significância</b>	<b>Os resultados obtidos devem agregar informação útil a avaliação da qualidade.</b>
<b>Custo e Complexidade de Aplicação</b>	O custo para definir e a complexidade de uma medida devem ser compatíveis com a avaliação a ser realizada.
<b>Repetibilidade</b>	As medidas devem ser repetíveis, exceto algumas que dificilmente podem ser repetidas por envolverem condições não controláveis.
<b>Reprodutabilidade</b>	Os resultados de uma avaliação devem ser os mesmos para avaliadores diferentes.
<b>Objetividade</b>	As opiniões de pessoas envolvidas devem ser limitadas ou totalmente evitadas.
<b>Imparcialidade</b>	As medidas devem ser imparciais, e deve se ter em mente que uma escolha não criteriosa de parâmetros pode levar a resultados diferentes e favorecer ou não um produto.
<b>Evidência para Validação</b>	As medidas devem prover evidências para sua validação. Por exemplo, se o resultado de uma avaliação é uma curva de tendência, é preciso prover informação a respeito do grau de confiabilidade dos valores apresentados.

- Levando em consideração o critério *Significância*, como mencionado no início desta seção, a coleta das medidas possibilitou melhorias presentes em novas versões dos sistemas avaliados. Com isso, é possível perceber a sua utilidade.
- Para o critério *Custo e Complexidade*, por ter feito uso da metodologia GQM, a participação dos especialistas durante as entrevistas foi de fundamental importância para que a complexidade de aplicação das medidas fosse a menor possível. Baseado nas experiências dos especialistas, o custo de definição também foi minimizado.
- Para o critério *Reprodutabilidade*, é importante destacar o padrão adotado para a definição das medidas, bem como, a presença explícita dos: momentos, procedimentos e periodicidade de coleta e análise. Permitindo, com isso, resultados iguais para avaliadores diferentes (quando avaliando o mesmo sistema sob as mesmas condições).
- Para satisfazer o critério *Objetividade*, as medidas foram definidas de forma que a sua coleta e interpretação não dão margem a opiniões subjetivas dos envolvidos (seja o desenvolvedor ou o usuário).
- Para alcançar o critério *Imparcialidade*, as medidas foram definidas seguindo um formato (apresentado no Capítulo 4) que impossibilita parcialidade por parte do avaliador no momento da coleta.
- E, por fim, para o critério *Evidência para Validação*, é importante destacar a elaboração do Guia de Medidas. Com ele, o avaliador poderá se basear nos valores de interpretação presentes no intervalo esperado dos dados possibilitando, com isso, evidências para a validação de seus resultados.

# Capítulo 6

## Conclusão

Este trabalho de dissertação apresentou um conjunto de medidas de qualidade de *software* para a avaliação da confiança no funcionamento (e.g., *dependability*) de sistemas ubíquos. Este capítulo apresenta na Seção 6.1 os resultados e contribuições alcançados e na Seção 6.2 os trabalhos futuros.

### 5.6 Resultados e Contribuições

Objetivando prover serviços em todo lugar e a qualquer momento de forma transparente, invisível e natural, os sistemas ubíquos são cada vez mais parte do nosso dia-a-dia. Para serem aceitos e utilizados, é aconselhável que esse tipo de sistema possibilite ao usuário um grau satisfatório de confiança no seu funcionamento. Para tanto, é necessário levar em consideração vários atributos de qualidade relacionados a confiança no funcionamento (e.g., *dependability*), tais como: disponibilidade, confiabilidade, proteção, confidencialidade, integridade e manutenibilidade.

É importante destacar também que a troca de informação entre as entidades envolvidas no sistema, o compartilhamento de recursos, a integração com serviços de terceiros e o uso de informações de contexto (e.g., localização e identificação do usuário) podem colocar em risco a segurança e privacidade do usuário nesses sistemas ubíquos, acarretando em problemas relacionados à confiança que ele tem no funcionamento do mesmo.

Diante desse cenário, uma das possíveis técnicas para a avaliação e garantia da confiança no funcionamento desses sistemas é a realização de medições (SCHOLTZ e CONSOLVO 2004), (SANTOS 2013). Porém, as diferenças (e.g., sensibilidade ao contexto, onipresença de serviços, entre outros) para os sistemas tradicionais indicam a necessidade de adequação dos métodos de avaliação existentes (e.g., medidas de qualidade).

Com estas motivações em mente, o objetivo desta pesquisa foi estabelecer um conjunto de medidas de qualidade de *software* para a avaliação da confiança no funcionamento de sistemas ubíquos com foco em aplicações para dispositivos móveis. Para tanto, diversas atividades foram

realizadas no desenvolvimento deste trabalho para alcançar os objetivos apresentados no Capítulo 1.

Para alcançar o objetivo (i) foi realizada uma revisão bibliográfica. Por meio da fundamentação teórica foi possível justificar a necessidade da adaptação de medidas existentes e definição de novas medidas para a avaliação da confiança no funcionamento em sistemas ubíquos, apoiando-se, principalmente, nas diferenças entre esse tipo de sistema e os sistemas tradicionais. Como mencionado no Capítulo 1, é fundamental destacar a importância do uso do mapeamento sistemático realizado por (SANTOS 2014), servindo este como base para diversos direcionamentos desta pesquisa durante a fase de revisão bibliográfica.

Ainda sobre a fundamentação teórica, os resultados identificados durante o levantamento bibliográfico expõem que as medidas existentes para os sistemas ubíquos não possuem, em sua maioria, uma definição completa (SANTOS 2014) (SCHOLTZ e CONSOLVO 2004) (RANGANATHAN 2005). Grande parte das medidas encontradas não apresenta funções de medição, valores de interpretação e métodos de coleta, inviabilizando uma avaliação de qualidade repetível e aplicável.

Para atingir o objetivo (ii), o método GQM foi utilizado. Como apresentado no Capítulo 2, essa metodologia consiste em: planejamento, definição, coleta e interpretação. Essas quatro etapas foram realizadas com sucesso e, para tanto, oito reuniões, com oito especialistas distintos, foram realizadas. Tais especialistas são professores, pesquisadores e estudantes da pós-graduação no grupo de pesquisa GREat (**G**rupos de pesquisa em **R**edes de computadores, **E**ngenharia de software e sistemas). No total são apresentadas neste trabalho vinte e seis medidas de qualidade de *software* que avaliam cinco questões de um objetivo de medição. Essas medidas foram aplicadas em três estudos de caso.

Os sistemas considerados para a realização dos estudos de caso foram: o *fAlert*, encontrado em (PIVA et al 2014), o *GREat Mute* apresentado em (SANTOS 2013), e o *GREat Tour*, um guia de visitas móveis sensível ao contexto, produto de uma Linha de Produto de Software Sensível ao Contexto chamada Moline, desenvolvida pelo grupo GREat e que pode ser encontrada em (MARINHO et al. 2012).

Com a realização dos estudos de caso foi possível alcançar os objetivos (iii) e (iv). A coleta das medidas permitiu a identificação de pontos de melhoria no que tange ao desenvolvimento dos três sistemas. Com isso, foi possível proporcionar um aumento significativo no grau de confiança no funcionamento e, conseqüentemente, na aceitação dos sistemas avaliados. Isso revela assim, indícios de que as medidas propostas são capazes de avaliar características de qualidade.

Diante deste cenário, este trabalho gerou como contribuição para medições de qualidade de *software* (ISO 25010), em particular de sistemas ubíquos: um conjunto de medidas de qualidade de *software* para a avaliação da confiança no funcionamento de sistemas ubíquos voltados para dispositivos móveis. Essas medidas foram definidas de forma a padronizar a sua nomenclatura, função de medição e métodos de coleta, e, para tanto, as medidas são acompanhadas da descrição dos procedimentos de coleta, análise e interpretação.

Dentro do grupo de pesquisa (GREat) no qual este trabalho está vinculado, esta dissertação contribui para o projeto Maximum, pois estabelece medidas de qualidade para a avaliação da confiança no funcionamento de sistemas ubíquos, e contribui também para o trabalho de (SANTOS 2014), pois define medidas para a característica confiança no funcionamento que se encaixam dentro das características encontradas pela autora em seu mapeamento sistemático (e.g., disponibilidade, integridade, confiabilidade, proteção).

A contribuição para o projeto Maximum é alcançada, pois, o conjunto de medidas proposto neste trabalho está diretamente associado a um dos objetivos específicos do projeto, o estabelecimento de medidas de qualidade para avaliação de sistemas ubíquos. A evolução do trabalho de (SANTOS 2013) é alcançada por meio da definição de medidas para características (e.g., disponibilidade, confiabilidade, proteção) encontradas no mapeamento sistemático realizado pelo autor, porém, não tratadas em seu trabalho. Por fim, é possível contribuir para a (ISO ISO 25010), pois, diferentemente da norma, ao avaliar atributos como disponibilidade, integridade e confiabilidade as medidas definidas neste trabalho levam em consideração informações obtidas em tempo de execução.

Com os resultados alcançados, apresentados anteriormente, foi possível a escrita, submissão e aceitação dos seguintes trabalhos científicos:

- “Uma Proposta de Medidas de Qualidade para Avaliação da Confiança em Sistemas Ubíquos” **Ferreira, A. B.**, Braga, R. B., Andrade, R. M. C. (aceito no Workshop de Teses e Dissertações em Qualidade de Software do XIII Simpósio Brasileiro de Qualidade de Software (WTDQS/SBQS 2014))

Este trabalho foi apresentado ainda no primeiro ano do mestrado, em um evento específico da área de qualidade de software. Nesse primeiro momento, houve a oportunidade de ouvir de especialistas sobre a relevância em fazer pesquisa na área de medições e dos desafios mediante o paradigma da computação ubíqua.

Mesmo estando no início da pesquisa, foi possível apresentar detalhes parciais sobre as medidas e os estudos de caso que seriam realizados e, obter bons *feedbacks*.

- “Avaliação da Confiança em um Sistema *Android* para Detecção e Alerta de Quedas” **Ferreira, A. B.**, Piva, L. S., Braga, R. B., Andrade, R. M. C. (aceito no XX Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2014))

Apresentado após a defesa da proposta de dissertação, pré-requisito do programa de mestrado. Neste trabalho, o segundo autor, aluno de graduação em intercâmbio no departamento de computação da UFC, gerou *logs* durante o desenvolvimento e testes do *fAlert* que foram utilizados para a coleta das medidas já documentadas. Este trabalho foi fundamental como fruto inicial dos resultados alcançados nesta etapa do mestrado e este artigo foi indicado a Best-Paper na categoria Short Paper na Trilha Principal e recebeu o prêmio de Menção Honrosa. Por ser um artigo curto, apenas 4 páginas, diversos resultados não foram apresentados. Servindo estes para a concepção de outro artigo, apresentado posteriormente.

- “*fAlert*: Um Sistema *Android* para Monitoramento de Quedas em Pessoas com Cuidados Especiais” Piva, L. S., **Ferreira, A. B.**, Braga, R. B., Andrade, R. M. C. (aceito no Workshop de Ferramentas e Aplicações do XX Simpósio Brasileiro de Sistemas Multimídia e Web - WFA/WebMedia 2014)

A aplicação *fAlert*, concebida com o intuito de ser um dos estudos de caso desta dissertação, também originou um artigo. O feedback dado pela banca presente no evento, mesmo não sendo sobre as medidas de qualidade desta pesquisa, serviu para diversos aspectos relacionados a concepção deste trabalho e futuras contribuições.

- “Avaliação da *Confiança no Funcionamento* de Sistemas de Detecção e Alerta de Quedas” **Ferreira, A. B.**, Braga, R. B., Andrade, R. M. C. (aceito no VII Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP 2015))

Este artigo é uma evolução do *short paper* aceito no WebMedia 2014. No primeiro, foram apresentados resultados de apenas duas medidas. Nesse segundo, seis medidas foram coletadas e interpretadas, gerando um volume mais significativo de resultados.

## 5.7 Trabalhos Futuros

As medidas de qualidade de *software* apresentadas neste trabalho possibilitam a avaliação da confiança no funcionamento de sistemas ubíquos para dispositivos móveis. Conseqüentemente, essas medidas viabilizam também a aceitação e utilização desse tipo de

sistema. Entretanto, ainda é preciso investigar desafios que ficaram em aberto, gerando as seguintes perspectivas de trabalhos futuros:

- Realizar medições em outros sistemas ubíquos para dispositivos móveis;

Por limitações relacionadas à pesquisa (e.g., tempo disponível, ausência de aplicações ubíquas de código-aberto, complexidade para desenvolver aplicações ubíquas críticas), não foi possível realizar todos os estudos de caso com sistemas onde a confiança no funcionamento seja considerada um atributo crítico. Sendo assim, é necessária a realização de novos estudos de caso em sistemas onde a confiança no funcionamento seja um atributo crucial.

- Analisar se o conjunto de medidas apresentado pode ser utilizado para a avaliação da confiança no funcionamento em outros tipos de sistemas ubíquos;

O conjunto de medidas apresentado neste trabalho auxilia a avaliação da confiança no funcionamento, porém, como foram definidas e avaliadas apenas em cenário de sistemas ubíquos que executam em dispositivos móveis, não há indícios de que possam ser aplicadas em outros domínios, acarretando na necessidade de novas avaliações e, possivelmente, novas medidas.

- Coletar as medidas em projetos reais de desenvolvimento de software;

As medidas foram aplicadas em sistemas desenvolvidos em projetos de pesquisa, não refletindo detalhes particulares e eventuais do processo de desenvolvimento de software. Todos os sistemas avaliados nos estudos de caso são provas de conceitos (PoCs, do inglês *proof of concept*). As PoCs avaliadas servem como modelo prático para confirmar os conceitos teóricos estabelecidos nesta pesquisa, porém não é possível fazer afirmações sobre a qualidade das medidas quando coletadas em produtos reais.

- Realizar um estudo mais aprofundado sobre a característica de confiança no funcionamento (e.g., dependability) e confiança (e.g., trust) e, eventualmente integrar no modelo de características apresentado em (SANTOS et al 2013);

- Analisar a dependência entre as medidas;

Assim como em (SANTOS 2014), é necessário verificar se o resultado de uma medida pode impactar ou complementar o resultado de outra, indicando assim uma dependência entre elas.

# Referências Bibliográficas

- Abi-Char, P., Mhamed, A., El-Hassan, B. and Mokhtari, M. (2010). “A Flexible Privacy and Trust Based Context-Aware Secure Framework”. In Lecture Notes in Computer Science.
- Abowd, G. D., Mynatt, E. D. and Rodden, T. (2002). “The Human Experience [of Ubiquitous Computing]”. *Pervasive Computing*, IEEE, v. 1, n. 1, p. 48–57.
- Alvez M. e Vasques F., “Confiança no Funcionamento: Conceitos Básicos e Terminologia – Relatório Técnico”, Portugal, 1998
- Araujo, R. B. De (2003). “Computação Ubíqua: Princípios, Tecnologias e Desafios”. In XXI Simpósio Brasileiro de Redes de Computadores.
- Avizienis, A., Vytautas Magnus Univ., Kaunas, Lithuania, Laprie, J.-C. ; Randell, B., Landwehr, C.; Basic concepts and taxonomy of dependable and secure computing, *IEEE Transactions on Dependable and Secure Computing*, 2004.
- Barbosa, S. D. J. and Silva, B. S. (2010). *Interação Humano-Computador*. Elsevier.
- Basili V., Caldiera G.. *The Goal Question Metric Approach*, 2002.
- Berard, E., Metrics for Object-Oriented Software Engineering, an Internet posting on comp.software-eng, janeiro de 1995.
- Bringel Filho, M. J. R., “CxtBAC : Une Famille de Modèles de Contrôle d'accès Sensible au Context e pour les Environnements Pervasifs”, Tese de Doutorado – Universidade de Grenoble, 2006.
- Bublitz, F. M., *Infra-estrutura para o Desenvolvimento de Aplicações Cientes de Contexto em Ambientes Pervasivos*, Campina Grande, 2007
- Chen, G.; Kotz, D. A Survey of Context-Aware Mobile Computing Research. Dartmouth Computer Science Technical Report TR2000-381, Department of Computer Science, Dartmouth College, New Hampshire, USA, 2000.
- Cheng, B. H. C. et al. *Software Engineering for Self-Adaptive Systems: A Research Roadmap*. LNCS, p. 1–26, 2009.

- Coulouris, G.; DOLLIMORE, Jean; KINDBERG, Tim. Distributed Systems: Concepts and Design, 5ª Edição, Addison-Wesley, 2011.
- D. A. Cooper and K. P. Birman, "Preserving Privacy in a Network of Mobile Computers," IEEE Symposium on Research in Security and Privacy, 1995.
- Davis, A. et al., Identifying and Measuring Quality in a Software Requirements Specification, Proc. 1st Intl. Software Metric Symposium, IEEE, Baltimore, MD, 2010, pg. 141-152
- Debashis S., Amitava M., "Pervasive Computing: A Paradigm for the 21st Century," Computer, pp. 25-31, March, 2003
- Dey A. K. Understanding and Using Context. Personal and Ubiquitous Computing, 5(1):4-7, 2001.
- Fogg, B. J., Tseng, H. (1999). The Elements of Computer Credibility, ACM CHI 99 Human Factors in Computing Systems Conference, 15-20.
- Ferreira, A. B., Braga, R. B., Andrade, R. M. C. , Avaliação da Confiança no Funcionamento de Sistemas de Detecção e Alerta de Quedas, VII Simpósio Brasileiro de Computação Ubíqua e Pervasiva, SBCUP 2015.
- Fudickar S., Karth C., Mahr P., Schnor B., "Fall-detection Simulator for accelerometers with in-hardware preprocessing", PETRA, Greece, 2012
- Goillau P., Kelly C., Boardman M., e Jeannot E.. (2003). Guidelines for Trust in Future ATM Systems: Measures. Document HRS/HSP-005-GUI-02. Brussels, Belgium. European Organisation for the Safety of Air Navigation.
- Gwizdka J. What's in the context. In Proceedings of the Computer Human Interaction 2000 (CHI2000), Workshop on "The What, Who, Where, When, Why and How of Context-Awareness", Hague, Holanda, 2000.
- Gu T., H. K. Pung, and D. Q. Zhang. A Bayesian Approach for Dealing with Uncertain Contexts. In G. Kotsis, editor, Proceedings of the 2nd International Conference on Pervasive Computing, volume 176, Viena, Áustria, 2004. Austrian Computer Society.
- Hamman R., Google Glass: saiba tudo sobre os óculos da Google, Disponível em: <<http://www.tecmundo.com.br/project-glass/25785-google-glass-saiba-tudo-sobre-osoculos-da-google.htm#ixzz2zzLjAvLT>> Acessado em: 25 de Abril de 2014

Hughes, B. (2000). Practical Software Measurement. McGraw - Hill Maidenhead.

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

ISO/IEC 9126 (2001).“Software Engineering – Product Quality – Part 1”.

Gresse, C., Hoisl, B., Wüst, J., A process model for planning GQM-based Measurement, Technical Report STTI-95-04-E, Software Technology Transfer Initiative, University of Kaiserslautern, October 1995.

Jia, L., Collins, M. and Nixon, P. (2009).“Evaluating Trust-Based Access Control for Social Interaction”.In 3rd International Conference on Mobile Ubiquitous Computing, Systems, Services, and Technologies, UBICOMM 2009. . Ieee.

Karvonen H., Different aspects of trust in ubiquitous intelligent transportation systems Proceeding ECCE '10 Proceedings of the 28th Annual European Conference on Cognitive Ergonomics Pages 311-314

Kemp, E. A., Johnson, R. S. and Thompson, A.-J. (2008). “Interface Evaluation for Invisibility and Ubiquity: An Example from E-learning”. In 9th ACM SIGCHI New Zealand Chapter’s International Conference on Human-Computer Interaction: Design Centered HCI. . ACM.

Kilinc C., Booth T., and Andersson K.. WallDroid: Cloud Assisted Virtualized Application Specific Firewalls for the Android OS. In Proceedings of the 11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom 2012), pages 877–883, 2012.

Koscianski, A. and Soares, M. dos S. (2007). “Qualidade de software”. São Paulo: Novatec

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

Langeheinrich M., Privacy by Design – Principles of Privacy Aware Ubiquitous Systems, in UBICOMP 2001, LNCS 2201, pp 273 291

Lee, J. and Yun, M. H. (2012).“Usability Assessment for Ubiquitous Services: Quantification of the Interactivity in Inter-Personal Services”.In IEEE International Conference on Management of Innovation & Technology (ICMIT). . Ieee.

- Leveson N. G. software Safety in Embedded Computer Systems. Communications of the ACM, Fevereiro de 1991, p. 35 - 45.
- Lima, E. R. R., Araújo, I. L., Santos, I. S., Oliveira, T. A., Monteiro, G. S., Costa, C. E. B., Santos, Z., Andrade, R. M. C. (2013). “GREat Tour: Um Guia de Visitas Móvel e Sensível ao Contexto”. In Webmedia 2013 - Workshop on Tools and Applications 2013.
- Lima, F. F. de P. (2011). “SysSu - Um Sistema de Suporte para Computação Ubíqua”. Dissertação de Mestrado - Universidade Federal do Ceará.
- Lima, F. F. P., Rocha, L. S., Maia, P. H. M. and Andrade, R. M. C. (2011). “Uma Arquitetura Desacoplada e Interoperável para Coordenação em Sistemas Ubíquos”. In Software Components, Architectures and Reuse (SBCARS).
- Liu Z., Shan J., Bonazzi R., Pigneur, Y., Date Privacy as a Tradeoff: Introducing the Notion of Privacy Calculus for Context-Aware Mobile Applications
- Loureiro E., Ferreira G., Almeida H., Perkusich A., Pervasive Computing: What Is It Anyway? Ubiquitous Computing: Design, Implementation and Usability. Information Science Reference, 2008, p. 9-36.
- Marinho, F. G.; Andrade, R. M. C.; Werner, C.; Viana, W.; Maia, M. E. F.; Rocha, L. S.; Teixeira, E.; Ferreira, J. B.; Dantas, V. L. L.; Lima, F.; Aguiar, S. (2012) “MobiLine: A Nested Software Product Line for the domain of mobile and contextaware applications”. Science of Computer Programming.
- Marinho, F. G., Costa, A. L., Lima, F. F. P., Neto, J. B B., Dantas, V. L. L., Andrade, R. M. C. (2010). “Uma Proposta de Arquitetura para Linhas de Produto de Software Aninhadas no Domínio de Aplicações Móveis e Sensíveis ao Contexto”. In IV Simpósio Brasileiro de Componentes, Arquiteturas e Reutilização de Software.
- Miranda, R. A. V., Provendo segurança e uma semântica de falhas consistente para a comunicação de objetos assíncronos distribuídos, Campina Grande, 2010.
- Nielsen, J. Usability Engineering. Academic Press, 1993.
- Nixon P., Lacey G. and Dobson S. (eds), Managing interactions in smart environments, Springer Verlag Press, pp. 243, December 1999
- Nixon P A , W. Wagealla, C. English, S. Terzis, Security, Privacy and Trust Issues in Smart Environments, The Global and Pervasive Computing Group Department of Computer and Information Sciences University of Strathclyde, Glasgow, Scotland, 2005.

- Noury N., Fleury A., Rumeau P., Bourke A. K., Rialle V., Lundy J. E., “Fall detection – Principles and Methods”, Proceedings of the 29th Annual International, 2007.
- Park, R. E., Goethert, W. B. and Florac, W. A. (1996). Goal-Driven Software Measurement. A Guidebook. Software Engineering Institute, Carnegie Mellon University.
- Peres, D. R. (2011). “Modelo de Qualidade para Componentes de Software”. Dissertação de Mestrado - Universidade de São Paulo.
- Petrell D., E. Not, C. Strapparava, O. Stock, and M. Zancanaro. Modeling Context is Like Taking Pictures. In Proceedings of the Computer Human Interaction 2000 (CHI2000), Workshop on “The What, Who, Where, When, Why and How of Context-Awareness”, Hague, Holanda, 2000.
- Piva, L. S., Ferreira, A. B., Braga, R. B., Andrade, R. M. C. , “fAlert: Um Sistema Android para Monitoramento de Quedas em Pessoas com Cuidados Especiais”, Workshop de Ferramentas e Aplicações do XX Simpósio Brasileiro de Sistemas Multimídia e Web, 2014
- Poppe, R., Rienks, R. and Dijk, B. Van (2007). “Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications”. AI for Human Computing, p. 234–250.
- Poslad, S. (2009). Ubiquitous Computing: Smart Devices, Environments and Interactions. Wiley.76
- Preece, J.; Rogers, Y.; Sharp H.; Benyon, D.; Holland, S., Carey T. (1994) Human Computer Interaction. Reading, MA., Addison-Wesley.
- Pressman, R. S. (2009). Software engineering: a practitioner’s approach. 7th. ed. McGraw-Hill.
- Ranganathan, A., Al-Muhtadi, J., Biehl, J., Ziebart, B., Campbell, R. H., Bailey, B. (2005). “Towards a Pervasive Computing Benchmark”. In Third IEEE International Conference on Pervasive Computing and Communications Workshops. . Ieee.
- Ranganathan, J. Al-Muhtadi, and RH. Campbell. Reasoning about Uncertain Contexts in Pervasive Computing Environments. IEEE Pervasive Computing, 3(2):62–70, 2004
- Rocha L.S., “CAEH: Um Método para Verificação de Modelos de Tratamento de Exceção Sensível ao Contexto em Sistemas Ubíquos”, Tese de Doutorado – Universidade Federal do Ceará, 2013.

- Rocha, L. S. (2010). “Dependabilidade em Sistemas Ubíquos”. Monografia de Qualificação- Universidade Federal do Ceará.
- Rocha, A. R. C., Souza, G. S., Barcellos M. C., “Medições de Software: Controle Estatístico de Processos”, PBQP Software, 2012.
- Rocha, L. S. et al. Ubiquitous Software Engineering: Achievements, Challenges and BeyondBrazilian Symposium on Software Engineering (in portuguese). Anais...set. 2011
- Ruhe G., Rough Set based Data Analysis in Goal Oriented Software Measurement. Third Symposium on Software Metrics, Berlin, March 1994, IEEE Computer Society Press, p.10-19
- Runeson, P.; Host, M. Guidelines for Conducting and Reporting Case Study Research in Software Engineering. Empirical Software Engineering, 19 dez. 2008.
- Santos R. M., Oliveira K. M., Andrade R. M. C., Santos I. S., Lima E. R.; “A Quality Model for Human-Computer Interaction Evaluation in Ubiquitous Systems” In: 6th Latin American Conference on Human Computer Interaction (CLIHC 2013).
- Santos, R. M. (2014). “Características e Medidas de Software para Avaliação da Qualidade da Interação Humano-Computador em Sistemas Ubíquos”. Dissertação de Mestrado - Universidade Federal do Ceará.
- Satyanarayanan M., Pervasive Computing: Vision and Challenges, IEEE Personal Communications , August 2001.
- Scalet, D., Koscianski, A., Vostoupal, T. M., Qualidade de software segundo normas internacionais. 1999.
- Schmidt, A. (jun 2000). “Implicit Human Computer Interaction Through Context”. Personal and Ubiquitous Computing, v. 4, n. 2, p. 191–199.
- Scholtz, J. and Consolvo, S. (2004). “Toward a Framework for Evaluating Ubiquitous Computing Applications”.IEEE Pervasive Computing.
- Sherchan W., Nepal S., Paris C., A survey of trust in social networks, Published in Journal ACM Computing Surveys (CSUR) 2014
- Sommerville, I. (2007) Software Engineering, 7th edition. Addison-Wesley

- Söllner, M., Hoffmann, A., Hoffmann, H., & Leimeister, J. M. (2011, December). Towards a Theory of Explanation and Prediction for the Formation of Trust in IT Artifacts. In Annual Workshop on HCI Research in MIS, Shanghai (Vol. 4).
- Söllner, M., Hoffmann, A., Hoffmann, H., & Leimeister, J. M. (2012, May). How to use behavioral research insights on trust for HCI system design. In CHI'12 Extended Abstracts on Human Factors in Computing Systems (pp. 1703-1708). ACM.
- Soligen, V. R. e Berghout, E., “The goal/question/metric method – A practical guide for quality improvement of software development”. Great Britain: Cambridge, McGraw-Hill, 1999
- Sousa, B., Pentikousis, K. and Curado, M. (2011). “UEF: Ubiquity Evaluation Framework”. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), v. 6649 LNCS, p. 92–103.
- Spínola, R. O., Massollar, J. and Travassos, G. (2007). “Checklist to Characterize Ubiquitous Software Projects”. Anais do Simpósio Brasileiro de Engenharia de Software, p. 39–55.
- Stajeno F., Security for Ubiquitous Computing, Wiley Press, 2002
- Stallings W. ,Network Security Essentials applications and standards, 2000.
- Sun, T. and Denko, M. K. (2008). “Performance Evaluation of Trust Management in Pervasive Computing”. In International Conference on Advanced Information Networking and Applications, AINA. . Ieee.
- Tinghuai M., Yong-Deak K., Qiang M., Meili T., and Weican Z. Context-aware implementation based on CBR for smart home. In Proceedings of IEEE International Conference on Wireless and Mobile Computing, Network and, volume 4, pages 112–115, Montreal, Canadá: IEEE Computer Society, 2005
- Treck D., The evaluation of Qualitative Assessment Dynamics (QAD) methodology for managing trust in pervasive computing environments, Published in: Pervasive Computing and Applications (ICPCA), 2011 6th International Conference on, 2011
- Weiser M., The Computer for the Twenty-First Century, Scientific American, pp. 94-10, September 1991.
- Viana, W. C. Mobilidade e sensibilidade ao contexto para a gestão de documentos multimídia pessoais: CoMMedia. Tese (Doutorado) — Université Joseph-Fourier - Grenoble, 2010. Disponível em: <<http://hal.archives-ouvertes.fr/tel-00499550/>>.

Yau, S. S., Wang, Y. and Karim, F. (2002). "Development of Situation-Aware Application Software for Ubiquitous Computing Environments". In Computer Software and Applications Conference.

# Apêndice A – Guia de Medidas

Como mencionado no Capítulo 2, as medidas de qualidade de software devem ser definidas de forma a tornar a atividade de avaliação algo repetível e aplicável. Para tanto, segundo (ROCHA et al 2012), é necessário levar em consideração diversas informações (e.g. descrição, função de medição, momento da coleta, periodicidade da coleta, responsável pela coleta).

Neste guia, as medidas definidas neste trabalho, seguem um formato baseado em (ROCHA et al 2012), contendo, assim como apresentado no Capítulo 4:

- Nome: nome da medida;
- Definição: descrição sucinta da medida;
- Mnemônico: sigla para identificar a medida, identificador;
- Tipo de Medida: classificação da medida quanto a sua dependência funcional, podendo ser uma medida base ou derivada;
- Entidade Medida: entidade que a medida mede;
- Propriedade Medida: propriedade da entidade medida que é quantificada pela medida;
- Unidade da Medida: unidade de medida em relação à qual a medida é medida;
- Tipo de Escala: natureza dos valores que podem ser atribuídos à medida;
- Valores da Escala: valores que podem ser atribuídos à medida;
- Intervalo Esperado dos Dados: limite de valores da escala definida;
- Fórmula de Medição: fórmula utilizada no procedimento de medição;
- Procedimento de Coleta: descrição do procedimento que deve ser realizado para coletar uma medida;
- Momento da Coleta: momento no qual a medida deve ser coletada;
- Periodicidade da Coleta: frequência de coleta da medida;
- Responsável pela Coleta: papel desempenhado pelo recurso humano responsável pela coleta da medida;
- Procedimento de Análise: descrição do procedimento que deve ser realizado para analisar uma medida;
- Momento da Análise: momento no qual a medida deve ser analisada;
- Periodicidade da Análise: frequência de análise da coleta da medida;
- Responsável pela Análise: papel desempenhado pelo recurso humano responsável pela análise dos resultados.

Além disso, para cada medida são descritos também o seu objetivo e a principal questão (GQM) relacionada. É importante ressaltar que uma mesma medida pode fornecer informações significativas para mais de uma questão, porém, neste guia, assim como em (ISO 25010), no momento de sua descrição formal, cada medida é associada apenas a uma pergunta.

As perguntas derivadas do GQM são:

- Qual o grau de disponibilidade do sistema?
  - Busca identificar se o sistema está disponível sempre que requisitado.
- Qual o grau de integridade do sistema?
  - Busca identificar a ausência de alterações inadequadas no sistema
- Qual o grau de confiabilidade do sistema?
  - Busca avaliar se o sistema funciona conforme o especificado.
- Qual o grau de manutenibilidade do sistema?
  - Busca avaliar o grau de eficácia e eficiência com o qual o sistema pode ser modificado.
- Qual o grau de proteção do sistema?
  - Busca avaliar o grau em que o sistema pode levar a um estado no qual a vida humana, a saúde, a propriedade ou o ambiente está em perigo
- Qual o grau de confidencialidade do sistema?
  - Busca avaliar a proteção da informação ou serviço contra acessos não autorizados

---

**Objetivo:** Maximizar a capacidade do sistema de prover serviços sempre que requisitado.

**Questão:** Qual o grau de disponibilidade do sistema?

---

*Nome da Medida: Grau de Disponibilidade*

*Definição: Capacidade de prestar serviço quando requisitado.*

*Mnemônico: M01*

*Tipo de Medida: Medida Base*

*Entidade Medida: Recurso de Software*

*Propriedade Medida: Disponibilidade (availability)*

*Unidade de Medida: Porcentagem*

*Tipo de Escala: Escala ordinal*

*Valores da Escala: Números reais positivos*

*Intervalo Esperado dos Dados: [0,1], quanto mais próximo de 1, melhor.*

*Função de Medição:*

$$\frac{TmF}{TmF+TmR}$$

*Onde TmF é o tempo médio entre falhas e TmR é o tempo médio para reparo*

*Procedimento de Coleta: Calcular a razão entre o tempo médio entre falhas e o tempo médio para reparo utilizando a função de medição apresentada acima. O cálculo pode ser feito de maneira automática com a geração de logs de falhas e ferramentas de apoio à medição.*

*Momento da Coleta: Durante atividades de testes relativos aos requisitos não funcionais.*

*Periodicidade da Coleta: Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional disponibilidade.*

*Responsável pela Coleta: Desenvolvedor ou Analista de Teste*

*Procedimento de Análise: Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Minimizar a ocorrência de erros durante o uso do serviço.

**Questão:** Qual o grau de disponibilidade do sistema?

---

**Nome da Medida:** Taxa de Erros

**Definição:** Taxa de ocorrência de erros durante o uso do serviço. Onde erro indica que algo inesperado aconteceu e a aplicação ou sistema parou de funcionar (e.g., travou, fechou sozinha). Não dando continuidade na prestação do serviço.

**Mnemônico:** M02

**Tipo de Medida:** Medida Base

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Disponibilidade (availability)

**Unidade de Medida:** Erros/Tempo

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:**  $[0, \infty]$ , quanto mais próximo de 0, melhor.

**Função de Medição:**

$$\frac{QtE}{TU}$$

Onde QtE é a quantidade de erros e TU é o tempo de uso

**Procedimento de Coleta:** Calcular a razão entre a quantidade de erros ocorridos e o tempo de uso utilizando a função de medição apresentada acima. O cálculo pode ser feito de maneira automática com a geração de logs de erros e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional disponibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.

**Responsável pela Análise:** Analista de qualidade.

---

---

**Objetivo:** Minimizar a ocorrência de falsos positivos decorrentes de uma percepção incorreta do contexto do usuário.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Precisão

**Definição:** Representa a fração de contextos sentidos ou inferidos que são relevantes para o sistema.

**Mnemônico:** M03

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Porcentagem

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:** [0,1] quanto mais próximo de 1, melhor.

**Função de Medição:** 
$$\frac{P}{P+FP}$$
  
Onde P representa um positivo e FP representa um falso positivo

**Procedimento de Coleta:** Calcular a razão entre as mudanças de contexto (relevantes para o provimento de serviço) que houve e o sistema detectou e todas as mudanças de contexto detectadas pelo sistema, sejam elas relevantes ou não. O cálculo pode ser feito de maneira automática com a geração de logs de erros e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.  
**Responsável pela Análise:** Analista de qualidade.

---

---

**Objetivo:** Minimizar a ocorrência de falsos negativos decorrentes de uma percepção incorreta do contexto do usuário.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Recall (abrangência ou cobertura)

**Definição:** Representa a fração de contextos relevantes para o sistema que foram corretamente sentidos ou inferidos.

**Mnemônico:** M04

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Porcentagem

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:** [0,1] quanto mais próximo de 1, melhor.

**Função de Medição:** 
$$\frac{P}{P+FN}$$
  
Onde P representa um positivo e FN representa um falso negativo

**Procedimento de Coleta:** Calcular a razão entre as mudanças de contexto (relevantes para o provimento de serviço) que houve e o sistema detectou e todas as mudanças de contexto relevantes para o sistema, tenham sido elas detectadas ou não. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.

**Responsável pela Análise:** Analista de qualidade.

---

---

**Objetivo:** Minimizar a ocorrência de falsos negativos decorrentes de uma percepção incorreta do contexto do usuário. A ideia-chave dessa medida é que as mudanças de contexto que não são relevantes para o provimento de serviços e o total de mudanças não relevantes identificados seja compatível.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Especificidade

**Definição:** Representa a fração de contextos não relevantes identificados pelo sistema sobre o total de contextos não relevantes

**Mnemônico:** M05

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Porcentagem

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:** [0,1] quanto mais próximo de 1, melhor.

**Função de Medição:** 
$$\frac{N}{N+FP}$$
  
Onde N representa um negativo e FP representa um falso positivo

**Procedimento de Coleta:** Calcular a razão entre as mudanças de contexto não relevantes para o provimento de serviço que houve e o sistema detectou corretamente (não reagindo) e todas as mudanças de contexto não relevantes para o sistema, tenham sido elas detectadas ou não. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

---

---

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

Objetivo: Minimizar a ocorrência de falsos positivos e negativos decorrentes de uma percepção incorreta do contexto do usuário.

---

Questão: Qual o grau de confiabilidade do sistema?

---

Nome da Medida: Grau de Desempenho (*f-Measure*)

*Definição: Representa a média ponderada entre o grau de precisão e o grau de recall. Em estatística a média é o valor que aponta para onde mais se concentram os dados de uma distribuição. Pode ser considerado o ponto de equilíbrio. Na média simples todos os valores possuem um mesmo peso, situação diferente na média ponderada, que para cada valor deve-se levar em conta o valor do seu peso.*

*Mnemônico: M06*

*Tipo de Medida: Medida Derivada*

*Entidade Medida: Recurso de Software*

*Propriedade Medida: Confiabilidade (reliability)*

*Unidade de Medida: Porcentagem*

*Tipo de Escala: Escala ordinal*

*Valores da Escala: Números reais positivos*

*Intervalo Esperado dos Dados: [0,1] quanto mais próximo de 1, melhor.*

*Função de Medição:* 
$$2 * \left( \frac{P}{P+FP} * \frac{P}{P+FN} \right) \div \left( \frac{P}{P+FP} + \frac{P}{P+FN} \right)$$

*Onde P representa um positivo, N representa um negativo, FP representa um falso positivo e FN um falso negativo*

*Procedimento de Coleta: Calcular a média ponderada entre o grau de precisão e o grau de recall utilizando a função de medição apresentada acima. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.*

*Momento da Coleta: Durante atividades de testes relativos aos requisitos não funcionais.*

*Periodicidade da Coleta: Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.*

*Responsável pela Coleta: Desenvolvedor ou Analista de Teste*

*Procedimento de Análise: Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

---

---

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Minimizar a ocorrência de falsos negativos e falsos positivos decorrentes de uma percepção incorreta do contexto do usuário.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Correção ou Acurácia

**Definição:** Representa a fração de contextos corretamente sentidos ou inferidos

**Mnemônico:** M07

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Porcentagem

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:** [0,1] quanto mais próximo de 1, melhor.

**Função de Medição:** 
$$\frac{P+N}{P+N+FN+FP}$$

Onde *P* representa um positivo, *N* representa um negativo, *FP* representa um falso positivo e *FN* um falso negativo

**Procedimento de Coleta:** Calcular a razão entre as mudanças de contexto corretamente detectadas pelo sistema e todas as mudanças de contexto detectadas, sejam elas corretas ou não. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.

**Responsável pela Análise:** Analista de qualidade.

---

---

**Objetivo:** Garantir que o tempo no qual a informação e/ou o serviço é entregue ao usuário seja adequado, analisando, para isso, o tempo que o sistema demora para se adaptar a partir do instante em que a informação de contexto foi sentida ou inferida.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Tempo de Adaptação

**Definição:** Representa o tempo necessário para que a adaptação (resultado de uma mudança contextual) ocorra, buscando identificar inicializações precoces ou tardias do serviço provido pelo sistema.

**Mnemônico:** M08

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Tempo

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números reais positivos

**Intervalo Esperado dos Dados:**  $[0, \infty]$  quanto mais menor, melhor.

**Função de Medição:**

$$\sqrt{\frac{\sum_{i=1}^n (t_i - \bar{t})^2}{n-1}}$$

Onde  $\bar{t}$  é a média dos tempos

**Procedimento de Coleta:** Calcular o desvio padrão dos tempos que o sistema demorou em se adaptar uma quando ocorre mudanças de contexto. O desvio padrão é uma medida de dispersão que mede a variabilidade dos valores à volta da média. O valor mínimo do desvio padrão é 0 indicando que não há variabilidade, ou seja, que todos os valores são iguais à média. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação dos requisitos não funcionais confiabilidade e disponibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se o resultado destoa significativamente do esperado. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

---

---

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário no momento/instante de tempo adequado.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Adaptação

**Definição:** Busca identificar se o sistema se adaptada sempre que necessário. Implicando na ausência de notificações equivocadas ou do provimento de serviço baseado em informações desatualizadas.

**Mnemônico:** M09

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Porcentagem

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** Números naturais positivos

**Intervalo Esperado dos Dados:** [0%,100%] quanto mais próximo de 100%, melhor.

**Função de Medição:**

$$\frac{\sum_{i=1}^n \frac{A_i}{B_i} * 100}{n}$$

Onde  $A_i$  representa o número de vezes que o sistema se adaptou,  $B_i$  representa o número de vezes que foi requisitada a adaptação e  $n$  o número dos tipos das diferentes adaptações.

**Procedimento de Coleta:** Calcular a razão entre o número de vezes que o sistema se adaptou e o número de vezes que foi requisitada a adaptação. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação dos requisitos não funcionais confiabilidade ou disponibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa significativamente dos valores esperados ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.

---

---

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário de forma adequada.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** Grau de Corretude na Execução da Adaptação

**Definição:** Representa a execução correta dos mecanismos ou regras de adaptação implicando em um provimento de serviço conforme o esperado.

**Mnemônico:** M10

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Confiabilidade (reliability)

**Unidade de Medida:** Regras ou mecanismo de adaptação executados por mudança de contexto

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) Alto  
(2) Médio  
(3) Baixo

**Procedimento de Coleta:** A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:

- (1) Alto: quando há a execução correta das regras ou mecanismos de adaptação.
- (2) Médio: quando há a execução de mais de uma regra ou mecanismo de adaptação sobre o mesmo contexto, implicando em uma adaptação incorreta. Porém não inviabilizando o provimento do serviço.
- (3) Baixo: execução de mais de uma regra ou mecanismo de adaptação sobre o mesmo contexto, implicando em uma adaptação incorreta e inviabilizando o provimento do serviço.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes

---

---

*momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário no tempo adequado.

---

**Questão:** Qual o grau de disponibilidade do sistema?

---

**Nome da Medida:** Grau de Discrição de mecanismos de segurança

**Definição:** Tempo usado para interagir com o subsistema de segurança (por exemplo, autenticação) para auxiliar a tarefa principal.

**Mnemônico:** M11

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Disponibilidade (availability)

**Unidade de Medida:** Tempo

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) Alto  
(2) Médio  
(3) Baixo

**Procedimento de Coleta:** A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:

- (1) Alto: minutos
- (2) Médio: segundos
- (3) Baixo: milissegundos

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional disponibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

**Periodicidade da Análise:** Após atividades de testes relativos aos requisitos não funcionais.

**Responsável pela Análise:** Analista de qualidade.

---

---

**Objetivo:** Garantir que o usuário é ciente de quais dos seus dados pessoais o sistema faz uso para o provimento do serviço conforme o especificado.

---

**Questão:** Qual o grau de proteção do sistema?

---

**Nome da Medida:** *Grau de Compreensão do Uso de Dados*

**Definição:** *Mede a capacidade do usuário de entender quais dos seus dados ou informações pessoais são utilizados pelo sistema.*

**Mnemônico:** *M12*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Proteção (safety)*

**Unidade de Medida:** *Conhecimento de informações, por parte do usuário, que são utilizadas pelo sistema.*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo]*

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) *Alto*  
(2) *Médio*  
(3) *Baixo*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita por observação, acompanhada de respostas providas pelo usuário por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:*

(1) *Alto: o usuário é ciente de todas as suas informações que são utilizadas pelo sistema.*

(2) *Médio: o usuário é ciente de apenas algumas das suas informações que são utilizadas pelo sistema.*

(3) *Baixo: o usuário não é ciente de nenhuma das suas informações que são utilizadas pelo sistema.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional proteção.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Representar em um gráfico os dados coletados nos diferentes*

---

---

*momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que há recursos disponíveis ao usuário para o controle de suas informações pessoais.

---

**Questão:** Qual o grau de confidencialidade do sistema?

---

**Nome da Medida:** *Grau de Permissão no Uso de Dados*

**Definição:** *Representa a existência de recursos disponíveis ao usuário para controlar as permissões de uso sobre suas informações pessoais: privacidade do conteúdo, identidade, localização e outros.*

**Mnemônico:** *M13*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Confidencialidade (confidentiality)*

**Unidade de Medida:** *Quantidade de recursos disponíveis*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo], Quanto maior o número de recursos disponíveis mais próximo de 'Alto' o valor da medida, quanto menor, mais próximo de 'Baixo'.*

**Função de Medição:**

$$X = A, \text{ onde } A \text{ é}$$

- (1) *Alto*
- (2) *Médio*
- (3) *Baixo*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita por observação, acompanhada de respostas providas pelo usuário por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:*

*(1) Alto: o usuário tem controle sobre todos os seus dados que são utilizados pelo sistema e pode alterar as permissões a qualquer momento.*

*(2) Médio: existem recursos disponíveis para o usuário controlar e alterar a permissão de alguns dos seus dados pessoais que são utilizados pelo sistema.*

*(3) Baixo: Não existem recursos disponíveis para o usuário controlar o uso dos seus dados por parte do sistema.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à*

---

---

*avaliação do requisito não funcional proteção.*

*Responsável pela Coleta: Desenvolvedor ou Analista de Teste*

*Procedimento de Análise: Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

Objetivo: Garantir que o número de funções providas pelo sistema e o número de funções executadas e (ou) esperadas é compatível.

---

Questão: Qual o grau de confiabilidade do sistema?

---

*Nome da Medida: Grau de Concordância das Ações do Sistema*

*Definição: O número de funções providas pelo sistema deve ser compatível com o número de funções executadas e esperadas (e.g., ao entrar em um ambiente o usuário espera receber uma única informação, logo, o sistema não deve fornecer nenhuma ou mais de uma informação).*

*Mnemônico: M14*

*Tipo de Medida: Medida Derivada*

*Entidade Medida: Recurso de Software*

*Propriedade Medida: Confiabilidade (reliability)*

*Unidade de Medida: Quantidade de recursos disponíveis*

*Tipo de Escala: Escala ordinal*

*Valores da Escala: [0, 1]*

*Intervalo Esperado dos Dados: [0, 1], quanto mais próximo de 1, melhor.*

*Função de Medição:*

$$X = \frac{A}{B}$$

*Onde, A = Número de funções executadas e esperadas B = Número de funções providas pelo serviço ubíquo.*

*Procedimento de Coleta: Calcular a razão entre o número de funções executadas e (ou) esperadas e o número de funções providas, utilizando a função de medição apresentada acima. O cálculo pode ser feito de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual.*

*Momento da Coleta: Durante atividades de testes relativos aos requisitos não funcionais.*

*Periodicidade da Coleta: Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.*

*Responsável pela Coleta: Desenvolvedor ou Analista de Teste*

*Procedimento de Análise: Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia significativamente dos valores esperado ou dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

---

---

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

Objetivo: Garantir que o sistema é capaz de fornecer o serviço sem necessitar da intervenção do usuário.

---

Questão: Qual o grau de confiabilidade do sistema?

---

*Nome da Medida: Grau de independência ou autonomia do sistema*

*Definição: Mede a necessidade de requisições de ajuda, por parte do sistema, para confirmar o resultado de alguma saída e (ou) serviço.*

*Mnemônico: M15*

*Tipo de Medida: Medida Derivada*

*Entidade Medida: Recurso de Software*

*Propriedade Medida: Confiabilidade (reliability)*

*Unidade de Medida: Quantidade de requisições de ajuda*

*Tipo de Escala: Escala ordinal*

*Valores da Escala: [Alto, Médio, Baixo]*

*Intervalo Esperado dos Dados: [Alto, Médio, Baixo], Quanto menor o número de requisições de ajuda mais próximo de 'Alto' o valor da medida, quanto maior, mais próximo de 'Baixo'.*

*Função de Medição:*

$X = A$ , onde  $A$  é

- (1) Alto
- (2) Médio
- (3) Baixo

*Procedimento de Coleta: A coleta pode ser feita de maneira automática com a geração de logs e ferramentas de apoio à medição ou, por observação e cálculo manual, acompanhada de respostas providas pelo usuário por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:*

*(1)Alto: o sistema não necessita ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço.*

*(2)Médio: o sistema necessita da ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço, porém as requisições de ajuda estão dentro do previsto na especificação do sistema e fazem parte de um conjunto de requisições de ajuda previamente esperadas/configuradas.*

*(3)Baixo: o sistema necessita da ajuda do usuário para confirmar o resultado de alguma saída e/ou serviço em situações onde esta requisição de ajuda não está previamente configurada/esperada.*

*Momento da Coleta: Durante atividades de testes relativos aos requisitos não funcionais.*

*Periodicidade da Coleta: Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.*

---

---

*Responsável pela Coleta: Desenvolvedor ou Analista de Teste*

*Procedimento de Análise: Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o sistema possui regras e diretrizes com intuito de proteger as informações e os ativos (algo que influi direto ou indiretamente), definindo normas ou técnicas para assegurar a integridade, disponibilidade e confidencialidade da informação.

---

**Questão:** Qual o grau de integridade do sistema?

---

**Nome da Medida:** *Presença de Políticas de Segurança*

**Definição:** *Representa a existência de políticas de segurança utilizadas pelo sistema ou pela aplicação.*

**Mnemônico:** *M16*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Integridade (integrity)*

**Unidade de Medida:** *Tipo de política de segurança (um número natural que representa o tipo de política de segurança)*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[0, 1, 2]*

**Intervalo Esperado dos Dados:** *[0, 1, 2]*

**Função de Medição:**

$X = A$ , onde  $A$  é

(0) *Nenhuma*

(1) *Presença de Política de Concessão Baseada em Permissão*

(2) *Presença de Política de Interação*

**Procedimento de Coleta:** *A coleta pode ser feita por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. De acordo com [Kilinc et al 2012], referência da qual foi retirada informações para a definição desta medida, as políticas de segurança em sistemas ubíquos são: Políticas de Concessão Baseada em Permissão e Políticas de Interação.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional integridade, disponibilidade ou proteção.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Analisar se há algum tipo de política de segurança implementada. Em caso negativo, conduzir uma investigação para avaliar se há a necessidade da implantação.*

**Momento da Análise:** *Durante atividades relativas à interpretação dos dados de coleta.*

**Periodicidade da Análise:** *Após atividades de testes relativos aos requisitos não funcionais.*

---

---

*Responsável pela Análise: Analista de qualidade.*

---

---

---

**Objetivo:** Garantir que o sistema desenvolvido possui documentação, auxiliando no seu entendimento e utilização.

---

**Questão:** Qual o grau de manutenibilidade do sistema?

---

**Nome da Medida:** Grau de Documentação

**Definição:** Representa a existência de documentação associada ao sistema desenvolvido.

**Mnemônico:** M17

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Manutenibilidade (maintainability)

**Unidade de Medida:** Documentação/Componente de Software

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**

$$X = A, \text{ onde } A \text{ é}$$

- (1) Alto
- (2) Médio
- (3) Baixo

**Procedimento de Coleta:** A coleta pode ser feita de maneira por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:

- (0) Alto: Todos os componentes estão documentados.
- (1) Média: Apenas uma parte dos componentes estão documentados.
- (2) Baixo: Não há documentação.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional manutenibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.

**Momento da Análise:** Durante atividades relativas à interpretação dos dados de coleta.

---

---

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o tempo necessário para a manutenção do sistema seja adequado.

**Questão:** Qual o grau de manutenibilidade do sistema?

---

**Nome da Medida:** Grau de Esforço para Realizar Manutenção do Sistema

**Definição:** Representa o esforço, relacionando-o ao tempo necessário para realizar manutenção do sistema.

**Mnemônico:** M18

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Manutenibilidade (maintainability)

**Unidade de Medida:** Tempo

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**

$X = A$ , onde  $A$  é  
(1) Alto  
(2) Médio  
(3) Baixo

**Procedimento de Coleta:** A coleta pode ser feita de maneira por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:

- (0) Alto: O tempo para realizar manutenção é superior ao tempo que foi necessário para implantação.
- (1) Média: O tempo para realizar manutenção é equivalente ao tempo que foi necessário para implantação.
- (2) Baixo: O tempo para realizar manutenção é inferior ao tempo que foi necessário para implantação.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional manutenibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam

---

---

*determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

---

**Objetivo:** Garantir que o tempo necessário para a manutenção do código seja adequado.

**Questão:** Qual o grau de manutenibilidade do sistema?

---

**Nome da Medida:** Grau de Esforço para Realizar Manutenção ou Evolução de Código

**Definição:** Representa o esforço, relacionando-o ao tempo necessário para realizar manutenção ou evolução do código

**Mnemônico:** M19

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Manutenibilidade (maintainability)

**Unidade de Medida:** Tempo

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**

$$X = A, \text{ onde } A \text{ é}$$

- (1) Alto
- (2) Médio
- (3) Baixo

**Procedimento de Coleta:** A coleta pode ser feita de maneira por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:

- (0) Alto: O tempo para realizar manutenção ou evolução é superior ao tempo que foi necessário para codificação.
- (1) Média: O tempo para realizar manutenção ou evolução é equivalente ao tempo que foi necessário para codificação.
- (2) Baixo: O tempo para realizar manutenção ou evolução é inferior ao tempo que foi necessário para codificação.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional manutenibilidade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam

---

---

*determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

---

**Objetivo:** Garantir que os dados transmitidos em rede não sejam acessados por quem não tem autorização.

---

**Questão:** Qual o grau de integridade do sistema?

---

**Nome da Medida:** Grau de Integridade dos Dados Transmitidos em Rede

**Definição:** Representa a existência da implementação de técnicas de integridade sobre os dados transmitidos em rede.

**Mnemônico:** M20

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Integridade (integrity)

**Unidade de Medida:** [Alto, Médio, Baixo]

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**

$$X = A, \text{ onde } A \text{ é}$$

- (1) Alto
- (2) Médio
- (3) Baixo

**Procedimento de Coleta:** A coleta pode ser feita de maneira por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:

- (0) Alto: Implementa técnicas de integridade com verificação e correção.
- (1) Média: Implementa técnicas de integridade, porém não garante verificação e correção.
- (2) Baixo: Não implementa técnicas de integridade.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional integridade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam

---

---

*determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

---

**Objetivo:** Garantir que os dados transmitidos em rede não sejam acessados por quem não tem autorização.

---

**Questão:** Qual o grau de integridade do sistema?

---

**Nome da Medida:** Grau de Integridade dos Dados Armazenados

**Definição:** Representa a existência da implementação de técnicas de integridade sobre os dados armazenados.

**Mnemônico:** M21

**Tipo de Medida:** Medida Derivada

**Entidade Medida:** Recurso de Software

**Propriedade Medida:** Integridade

**Unidade de Medida:** [Alto, Médio, Baixo]

**Tipo de Escala:** Escala ordinal

**Valores da Escala:** [Alto, Médio, Baixo]

**Intervalo Esperado dos Dados:** [Alto, Médio, Baixo]

**Função de Medição:**

$$X = A, \text{ onde } A \text{ é}$$

- (1) Alto
- (2) Médio
- (3) Baixo

**Procedimento de Coleta:** A coleta pode ser feita de maneira por observação e cálculo manual, acompanhada de respostas providas pelo desenvolvedor por meio de questionários. Para o cálculo da função de medição os seguintes aspectos devem ser levados em consideração:

- (0) Alto: Implementa técnicas de integridade com verificação e correção.
- (1) Média: Implementa técnicas de integridade, porém não garante verificação e correção.
- (2) Baixo: Não implementa técnicas de integridade.

**Momento da Coleta:** Durante atividades de testes relativos aos requisitos não funcionais.

**Periodicidade da Coleta:** Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional integridade.

**Responsável pela Coleta:** Desenvolvedor ou Analista de Teste

**Procedimento de Análise:** Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam

---

---

*determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado, mesmo quando o dispositivo estiver bloqueado (em descanso).

---

**Questão:** Qual o grau de disponibilidade do sistema?

---

**Nome da Medida:** *Grau de Funcionamento em Background*

**Definição:** *Avaliar se o sistema funciona em background.*

**Mnemônico:** *M22*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Disponibilidade (availability)*

**Unidade de Medida:** *[Alto, Médio, Baixo]*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo]*

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) *Alto*  
(2) *Médio*  
(3) *Baixo*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:*

- (1) Alto: o sistema funciona normalmente em background*
- (2) Médio: Algum aspecto do sistema é comprometido quando o dispositivo está bloqueado e a aplicação rodando, conseqüentemente, em background*
- (3) Baixo: O sistema não funciona em background*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade e disponibilidade.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

**Momento da Análise:** *Durante atividades relativas à interpretação dos dados de coleta.*

**Periodicidade da Análise:** *Após atividades de testes relativos aos requisitos não funcionais.*

---

---

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado, mesmo quando o dispositivo estiver com bateria baixa. A ideia-chave desta medida é avaliar como o sistema se comporta quando a bateria do dispositivo está baixa (menos de 25%). Algumas funções nativas do desenvolvimento para sistemas *Android* (o comando *broadcast*, por exemplo) não são ativadas (ou ativadas em intervalos de tempo maiores do que o adequado) quando o dispositivo está com pouca bateria.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** *Grau de Independência da Bateria*

**Definição:** *Avaliar se o sistema funciona com o dispositivo em diferentes níveis de bateria.*

**Mnemônico:** *M23*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Confiabilidade (reliability)*

**Unidade de Medida:** *[Alto, Médio, Baixo]*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo]*

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) *Alto*  
(2) *Médio*  
(3) *Baixo*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:*

(1) *Alto: o sistema funciona conforme o especificado e sempre que necessário em qualquer nível de bateria*

(2) *Médio: Algum aspecto do sistema é comprometido quando o dispositivo está com pouca bateria (menos de 25%)*

(3) *Baixo: O sistema não funciona quando o dispositivo está com pouca bateria.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade e disponibilidade.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

---

---

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado sem precisar comprometer significativamente a bateria do dispositivo.

---

**Questão:** Qual o grau de confiabilidade do sistema?

---

**Nome da Medida:** *Grau de Influência sobre a Bateria*

**Definição:** *Avaliar a influência do sistema sobre o consumo de bateria do dispositivo.*

**Mnemônico:** *M24*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Confiabilidade (reliability)*

**Unidade de Medida:** *[Alto, Médio, Baixo]*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo]*

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) *Baixo*  
(2) *Médio*  
(3) *Alto*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:*

- (1) *Baixo: a diferença entre o consumo de bateria com o sistema rodando ou sem rodar não ultrapassa 10%*
- (2) *Médio: a diferença entre o consumo de bateria com o sistema rodando ou sem rodar não ultrapassa 15%*
- (3) *Alto: a diferença entre o consumo de bateria com o sistema rodando ou sem rodar ultrapassa 15%.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoia dos demais. Em caso afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

**Momento da Análise:** *Durante atividades relativas à interpretação dos dados de coleta.*

---

---

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

---

**Objetivo:** Garantir que o serviço é entregue ao usuário conforme o especificado e sempre que requisitado independente de componentes externos. Por exemplo, um sistema que coloca o celular no modo de acordo com os compromissos da *Google Agenda* do usuário. Caso ocorra algum problema com o sistema de agendas da *google* por um tempo, o sistema presente no dispositivo, que interage com a agenda, também deixará de funcionar?

---

**Questão:** Qual o grau de disponibilidade do sistema?

---

**Nome da Medida:** *Grau de Independência de Componentes Externos*

**Definição:** *Avaliar se o sistema funciona conforme o especificado e sempre que requisitado independente de componentes externos.*

**Mnemônico:** *M25*

**Tipo de Medida:** *Medida Derivada*

**Entidade Medida:** *Recurso de Software*

**Propriedade Medida:** *Disponibilidade (availability)*

**Unidade de Medida:** *[Alto, Médio, Baixo]*

**Tipo de Escala:** *Escala ordinal*

**Valores da Escala:** *[Alto, Médio, Baixo]*

**Intervalo Esperado dos Dados:** *[Alto, Médio, Baixo]*

**Função de Medição:**  $X = A$ , onde  $A$  é  
(1) *Alto*  
(2) *Médio*  
(3) *Baixo*

**Procedimento de Coleta:** *A coleta desta medida pode ser feita de maneira automática com o uso de ferramentas de apoio à medição ou, por observação e cálculo manual. Para tanto, os seguintes aspectos devem ser levados em consideração:*

- (1) *Alto: O sistema funciona conforme o especificado e sempre que necessário mesmo que qualquer componente externo tenha algum problema.*
- (2) *Médio: Algum aspecto do sistema é comprometido quando algum componente externo, facilmente identificável, tem algum problema.*
- (3) *Baixo: O sistema não funciona quando qualquer componente externo tem algum problema.*

**Momento da Coleta:** *Durante atividades de testes relativos aos requisitos não funcionais.*

**Periodicidade da Coleta:** *Uma vez em cada ocorrência de atividade de teste relativa à avaliação do requisito não funcional confiabilidade e disponibilidade.*

**Responsável pela Coleta:** *Desenvolvedor ou Analista de Teste*

**Procedimento de Análise:** *Representar em um gráfico os dados coletados nos diferentes momentos de medição. Analisar se há algum valor que destoa dos demais. Em caso*

---

---

*afirmativo, conduzir uma investigação de causas para que ações corretivas sejam determinadas.*

*Momento da Análise: Durante atividades relativas à interpretação dos dados de coleta.*

*Periodicidade da Análise: Após atividades de testes relativos aos requisitos não funcionais.*

*Responsável pela Análise: Analista de qualidade.*

---

## Apêndice B – Formulário para traçar o perfil do usuário

O formulário a seguir foi utilizado antes da realização dos testes mencionados nos estudos de caso apresentados no Capítulo 5, com o intuito de traçar o perfil dos usuários envolvidos.

1. Nome
2. Sexo  
(1) Feminino (2) Masculino
3. Idade
4. Ocupação Profissional
5. Formação Acadêmica
6. Qual seu grau de experiência no uso de dispositivos móveis (e.g., smartphones, tablets)?  
(1)Muito Baixo (2)Baixo (3)Médio (4)Alto (5) Muito Alto
7. 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”
8. Qual sua experiência com aplicações ubíquas?  
(1)Muito Baixo (2)Baixo (3)Médio (4)Alto (5) Muito Alto
9. Você já desenvolveu softwares?  
(1)Sim (2)Não
10. Você já desenvolveu softwares ubíquos?  
(1)Sim (2)Não

## Apêndice C – Questionário utilizado para avaliação realizada pelo usuário

O conjunto de perguntas apresentadas neste questionário pode ser utilizado para a coleta de medidas que envolvem o uso de questionários (apontado como método de coleta na definição da medida, apresentado no Apêndice A) onde, o ponto de vista a ser avaliado é o do usuário final. Para as medidas que envolvem métodos de coleta automáticos, algumas das perguntas deste questionário podem servir como fonte de comparação para avaliação dos resultados obtidos. Algumas das perguntas deste questionário foram extraídas de (SANTOS 2014).

**Nome:**

**Data:**

- 1) A aplicação estava disponível quando você necessitou?
  - (1) A aplicação estava sempre disponível para interagir quando era necessário. Você conseguiu executar normalmente a função desejada sem nenhuma dificuldade.
  - (2) Geralmente estava disponível quando era necessário. Você conseguiu executar a aplicação, porém com dificuldades.
  - (4) A disponibilidade não corresponde com sua necessidade de onde e quando utilizar a aplicação. Você não foi capaz de executar funções desejadas.
  
- 2) Ocorreram erros durante o uso da aplicação (entenda por erro algo inesperado que fez com que a aplicação parasse de executar (e.g., travar ou encerrar sozinha))?
  - (1) Sim (2) Não
  
- 3) Quantos erros ocorreram durante o uso da aplicação?
  
- 4) O tempo de adaptação (ocorrência de uma mudança de contexto e a reação do sistema provendo o serviço foi):
  - (1) Minutos (2) Segundos (3) Milissegundos
  
- 5) Você considera esse tempo:
  - (1) Alto (2) Médio (3) Baixo
  
- 6) A adaptação foi correta (o serviço foi provido conforme o esperado)?

- 7) O tempo usado para interagir com o subsistema de segurança (por exemplo, autenticação) para auxiliar a tarefa principal foi:
- (1) Minutos (2) Segundos (3) Milissegundos
- 8) Você considera esse tempo:
- (2) Alto (2) Médio (3) Baixo
- 9) O usuário tem capacidade de entender quais dos seus dados ou informações pessoais são utilizados pelo sistema:
- (1) Alto: o usuário é ciente de todas as suas informações que são utilizadas pelo sistema.
  - (2) Médio: o usuário é ciente de apenas algumas das suas informações que são utilizadas pelo sistema.
  - (3) Baixo: o usuário não é ciente de nenhuma das suas informações que são utilizadas pelo sistema.
- 10) Existem recursos disponíveis ao usuário para controlar as permissões de uso sobre suas informações pessoais: privacidade do conteúdo, identidade, localização e outros:
- (1) Alto: o usuário tem controle sobre todos os seus dados que são utilizados pelo sistema e pode alterar as permissões a qualquer momento.
  - (2) Médio: existem recursos disponíveis para o usuário controlar e alterar a permissão de alguns dos seus dados pessoais que são utilizados pelo sistema.
  - (3) Baixo: Não existem recursos disponíveis para o usuário controlar
- 11) A aplicação fornece somente informações relevantes?
- (1) Nenhuma informação ou requisição irrelevante é apresentada. Você obteve somente informações e requisições necessárias.
  - (2) A maioria das informações e requisições são relevantes.
  - (3) Informações e requisições irrelevantes foram frequentemente apresentadas.
- 12) A aplicação interage no momento correto?
- (1) 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.
  - (2) 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.

(3) Na maioria das vezes, você não concordou com a interação.

13) Você notou diferenças no consumo de bateria após a instalação e uso da aplicação?

(2) Sim (2) Não

## Apêndice D – Questionário utilizado para avaliação realizada pelo desenvolvedor

O conjunto de perguntas apresentadas neste questionário pode ser utilizado para a coleta de medidas que envolvem o uso de questionários (apontado como método de coleta na definição da medida, apresentado no Apêndice A) onde, o ponto de vista a ser avaliado é o do desenvolvedor. Para as medidas que envolvem métodos de coleta automáticos, algumas das perguntas deste questionário podem servir como fonte de comparação para avaliação dos resultados obtidos. Algumas das perguntas deste questionário foram extraídas de (SANTOS 2014).

**Nome:**

**Data:**

1. A aplicação exige alguma intervenção explícita do usuário para receber o serviço?  
(1) Sim (2) Não
2. Quais intervenções o usuário pode ou precisa fazer?
3. Quais sensores do dispositivo são utilizados?
4. Para quais versões do *Android* a aplicação funciona?
5. Para quais dispositivos móveis a aplicação será executada (*tablet, smarphone*)?
6. Qual a quantidade de tecnologias de acesso a comunicação a aplicação tem acesso, descreva quais (*wifi, 3G*)?
7. A aplicação possui entradas implícitas (aquelas que não precisam ser efetuadas de maneira direta pelo usuário)?  
(1) Sim (2) Não
8. Qual a quantidade de entradas implícitas da aplicação?
9. Descreva as entradas implícitas que a aplicação utiliza.
10. Um usuário leigo da aplicação é capaz de perceber essas entradas implícitas?  
(1) Sim (2) Não
11. Quais entradas implícitas o usuário pode perceber?

12. A entrada implícita pode ser manipulada pelo o usuário?
- (1) Sim (2) Não
13. Quais entradas implícitas o usuário pode manipular?
14. Quantas e quais ações o usuário precisa fazer para a aplicação inicializar?
15. A aplicação possui interfaces para a interação natural como baseada em voz, gestos ou outros?
- (1) Sim (2) Não
16. Existe alguma política de segurança utilizada pela aplicação?
- (0) Nenhuma  
(1) Presença de Política de Concessão Baseada em Permissão  
(2) Presença de Política de Interação
17. Existe documentação associada ao sistema desenvolvido?
- (0) Alto: Todos os componentes estão documentados.  
(1) Média: Apenas uma parte dos componentes estão documentados.  
(2) Baixo: Não há documentação.
18. Qual o esforço, relacionando ao tempo necessário para realizar manutenção do sistema
- (0) Alto: O tempo para realizar manutenção é superior ao tempo que foi necessário para implantação.  
(1) Média: O tempo para realizar manutenção é equivalente ao tempo que foi necessário para implantação.  
(2) Baixo: O tempo para realizar manutenção é inferior ao tempo que foi necessário para implantação.
19. Qual o esforço relacionando ao tempo necessário para realizar manutenção ou evolução do código
- (0) Alto: O tempo para realizar manutenção ou evolução é superior ao tempo que foi necessário para codificação.  
(1) Média: O tempo para realizar manutenção ou evolução é equivalente ao tempo que foi necessário para codificação.  
(2) Baixo: O tempo para realizar manutenção ou evolução é inferior ao tempo que foi necessário para codificação.
20. Existem técnicas de integridade sobre os dados transmitidos em rede?
- (0) Alto: Implementa técnicas de integridade com verificação e correção.  
(1) Média: Implementa técnicas de integridade, porém não garante verificação e correção.  
(2) Baixo: Não implementa técnicas de integridade.

**21.** Existem técnicas de integridade sobre os dados armazenados?

(0) Alto: Implementa técnicas de integridade com verificação e correção.

(1) Média: Implementa técnicas de integridade, porém não garante verificação e correção.

(2) Baixo: Não implementa técnicas de integridade