

PODER EXECUTIVO MINISTÉRIO DA EDUCAÇÃO UNIVERSIDADE FEDERAL DO AMAZONAS INSTITUTO DE COMPUTAÇÃO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



Rodrigo dos Anjos Cruz Reis

MTCONTEXT: APOIO À GERAÇÃO DE CASOS DE TESTE PARA APLICAÇÕES MÓVEIS SENSÍVEIS AO CONTEXTO

RODRIGO DOS ANJOS CRUZ REIS

MTCONTEXT: APOIO À GERAÇÃO DE CASOS DE TESTE PARA APLICAÇÕES MÓVEIS SENSÍVEIS AO CONTEXTO

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas, como parte dos requisitos necessários para a obtenção do título de Mestre em Informática.

Manaus 2016

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

R375m

Reis, Rodrigo dos Anjos Cruz

MTContext: Apoio à geração de casos de teste para aplicações móveis sensíveis ao contexto. / Rodrigo dos Anjos Cruz Reis. 2016 103 f.: il. color; 31 cm.

Orientador: Arilo Claudio Dias Neto Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Sensibilidade ao Contexto. 2. Ubiquidade. 3. Teste de Software. 4. Aplicações Móveis. I. Dias Neto, Arilo Claudio II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO MINISTÉRIO DA EDUCAÇÃO INSTITUTO DE COMPUTAÇÃO



PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

FOLHA DE APROVAÇÃO

"MTCONTEXT: Apoio à Geração de Casos de Teste para Aplicações Móveis Sensíveis ao Contexto"

RODRIGO DOS ANJOS CRUZ REIS

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Arilo Claudio Dios Meto Prof. Arilo Claudio Dias Neto - PRESIDENTE

Profa. Elaine Harada Teixeira de Oliveira - MEMBRO INTERNO

Prof. Rodrigo Oliveira Spínola - MEMBRO EXTERNO

Manaus, 14 de Março de 2016



Agradecimentos

Tenho que agradecer a todos, pois este trabalho mesmo sendo uma monografia, com certeza não foi feito só por mim. São tantas pessoas envolvidas que precisaria de mais umas 100 páginas para citar cada uma. Sendo assim, me esforçarei para resumir e não esquecer de ninguém.

Primeiramente e obviamente agradeço a Deus pois se não fosse por ele, eu nem teria começado este mestrado. Minha saúde, minhas oportunidades, minhas forças nos momentos de aflição e segurança tudo veio dele e à Ele agradeço. Com Deus o impossível é só questão de Tempo e no tempo de Deus.

Agradeço à mulher que quando entrei no mestrado era minha namorada, durante o mestrado se tornou minha noiva e ao fim desse ciclo de mestrado é minha esposa. Jamilla Reis sempre dizia "eu sei que tudo que você faz, é pensando em nós". Sempre foi.

Um agradecimento mais que especial aos meus pais Francisco Reis que sempre vou lembrar do apoio dado ao dizer "qualquer falta de respeito é só chamar" e Rosineide dos Anjos que muitas vezes, acho que ela sentia que eu precisava dela, ela vinha de Boa Vista para Manaus sem nem pensar duas vezes para me incentivar também cobrar resultados. Mães. Tenho que citar também minha segunda mãe Elisia Cruz, quando digo que ela é minha segunda mãe é porque ela com certeza faz por merecer um título tão importante desses.

Agradeço a meus familiares de Boa Vista que me apoiaram e aos meus familiares de Manaus que também me apoiaram muita das vezes me recebendo em suas casas com todo amor e carinho. Esse apoio realmente eu nunca esquecerei, em especial aos meus primos Hewerton e Herywelton que estiveram sempre junto comigo durantes esses anos de pesquisa.

Aos meus amigos tantos os antigos quanto os que fiz durante esta pesquisa. Em especial a Thaynã Gonçalves que me aturou por praticamente dois anos morando comigo. Histórias nossas não vão faltar e fico muito feliz de poder viver mais essa contigo ao meu lado. Aos meus amigos que fiz durante as tão temidas disciplinas de PAA e FTC. Esses grupos de estudo fizeram toda a diferença. Não posso esquecer da Dhanielly Paulina que também teve o prazer da minha companhia durante alguns meses morando comigo.

Ao meu orientador Dias-Neto A. C. que foi paciente, atencioso e acreditou no menino novo sem muita experiência. Sem dúvidas o aprendizado que ganhei durante esses anos de convivência servirão para a vida toda.

É claro que agradecerei agora e sempre ao grupo de Experimentação e Teste de Software - ExperTS por ordem alfabética Bezerra, A. j. S.; Costa, E. A. B.; Lobão, L. M. A.;

Marques K. O.; Meireles, S. R. A.; Mendes, I. N.; Nascimento, J. P.; Neves, L. L.; Rodrigues, A. C. B.; Rojas, I. K. V.; e Trovão, J. C..

É preciso agradecer ao Raphael Rocha que foi uma das peças fundamentais desse trabalho. A Elizângela Santos, parceirona. E principalmente ao Grupo de boas Awdren de Lima Fontão, Jonathas Silva dos Santos e Renata Magalhães Rêgo meus parceiros de pesquisa, amigos e padrinhos de casamento. Não existem palavras no mundo que possam ser usadas para expressar minha gratidão e felicidade em ter vocês ao meu lado.

A todos os professores e funcionários do ICOMP que me guiaram e deram suporte durante esses anos de pesquisa.

Agradeço a CAPES, ao CNPq e ao INDT pelo suporte financeiro durante meus anos de pesquisa. Afinal, as bolsas me propiciaram momentos incríveis.

Por fim, quero agradecer a você que tirou um tempo para ler esta dissertação.

Resumo da Dissertação apresentada à UFAM/AM como parte dos requisitos necessários para a obtenção do grau de Mestre em Informática (M.Sc.)

MTCONTEXT: APOIO À GERAÇÃO DE CASOS DE TESTE PARA APLICAÇÕES MÓVEIS SENSÍVEIS AO CONTEXTO

Rodrigo dos Anjos Cruz Reis

Fevereiro / 2015

Orientador: Prof. Dr. Arilo Claudio Dias Neto

A computação móvel está atualmente em todos os lugares. Aplicações executando em dispositivos móveis (ex: smartphones e tablets) estão se tornando cada vez mais populares e elas representam uma revolução no setor de tecnologia da informação.

O exponencial crescimento deste mercado e da criticidade dos sistemas desenvolvidos impõem uma maior atenção a aspectos de confiabilidade destas aplicações. As aplicações móveis não são livres de defeitos, e novas abordagens de engenharia de software são necessárias para testar as aplicações.

Além do aumento considerável da utilidade de dispositivos móveis nos últimos anos, permitindo aos usuários executar mais tarefas em um contexto móvel, os avanços tecnológicos nas áreas de dispositivos móveis e redes de comunicação tem tornado possível a sensibilidade ao contexto, característica da computação ubíqua. Umas das formas de avaliar a qualidade dessas aplicações é por meio de testes.

Este trabalho propõe a definição de uma técnica de teste para apoiar a geração de casos de teste para aplicações móveis sensíveis ao contexto visando diminuir a quantidade de casos de testes gerados para este tipo de aplicação, diminuir o tempo de geração desses casos de teste, aumentar a cobertura de contextos alcançados com esses casos de teste e facilitar o gerenciamento dos testes gerados por meio de um apoio ferramental.

Os resultados obtidos no experimento controlado realizado com dois cenários de aplicações móveis sensíveis ao contexto forneceram indícios de viabilidade deste trabalho visto que a abordagem *MTContext* contribuiu para a geração de casos de teste com diferença estatisticamente significante em relação a abordagem de comparação Ad Hoc no que se diz respeito a maior cobertura de contextos. Porém, os resultados deste experimento evidenciaram limitações e possíveis evoluções para abordagem MTContext que serão tratados em trabalhos futuros.

Palavra Chave: Sensibilidade ao Contexto, Ubiquidade, Teste de Software, Aplicações Móveis.

Abstract of Thesis presented to UFAM/AM as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

MTCONTEXT: SUPPORT TO TEST CASE GENERATION FOR CONTEXT AWARE **MOBILE APPLICATIONS**

Rodrigo dos Anjos Cruz Reis

February / 2015

Advisor: Arilo Claudio Dias Neto

Mobile computing is currently everywhere. Applications running on mobile devices (e.g. smartphones and tablets) are becoming increasingly popular and they represent a revolution in the information technology sector.

The exponential growth of this market and the criticality of the developed systems impose greater attention to reliability aspects of these applications. Mobile applications are not free of defects, and new software engineering approaches are needed to test them.

In addition to the considerable increase in the use of mobile devices in recent years, allowing users to perform more tasks in a mobile environment, technological advances in mobile devices and communication networks, has made it possible the context awareness, characteristic of ubiquitous computing. One of the ways to evaluate the quality of these applications is through testing.

This paper proposes the definition of a testing technique to support the generation of test cases for Context Aware mobile applications in order to decrease the amount of test cases generated for this type of application, decrease the generation time of these test cases increase the contexts coverage achieved with these test cases and facilitate the management of the tests generated by a tool support.

The results obtained in controlled experiment conducted with two scenarios of context aware mobile applications provided evidence regarding the feasibility of this work. MTContext approach contributed to the generation of test cases with significant statistical difference when compared to Ad Hoc approach concerned with the higher contexts coverage during test cases generation. However, the results of this experiment showed limitations and possible upgrades to MTContext approach that will be addressed in future work.

Keywords: Context Awareness, Ubiquitous, Software test, Mobile Applications.

ΙX

SUMÁRIO

ÍNDICE DE	FIGURAS	XIV
ÍNDICE DE	TABELAS	XV
ÍNDICE DE	QUADROS	XVI
CAPÍTULO	1 - INTRODUÇÃO	17
1.1.	Contextualização e Motivação	17
1.2.	Descrição do Problema	19
1.3.	Hipótese	20
1.4.	Objetivos	21
1.4.1	Objetivo geral	21
1.4.2	Objetivos específicos	21
1.5.	Metodologia da Pesquisa	21
1.5.1	Fase de Concepção	22
1.5.2	Fase de Avaliação	22
1.6.	Estrutura do Documento	23
CAPÍTULO	2 - REFERENCIAL TEÓRICO	24
2.1.	Aplicações móveis	24
2.2.	Ubiquidade	25
2.3.	Aplicações Sensíveis ao Contexto	26
2.4.	Teste de Software	27
2.4.1	Teste de Aplicações móveis	28
2.4.2	Teste de Aplicações Móveis Sensíveis ao Contexto	29
2.5.	Trabalhos relacionados	30
2.5.1	. Geração de Teste para Aplicações Móveis Sensíveis ao Co	ontexto.31
2.6.	Considerações Finais	33
CAPÍTULO	3 - ABORDAGEM DE GERAÇÃO DE TESTE PARA APL	ICAÇÕES
MÓVEIS SE	NSÍVEIS AO CONTEXTO	34
3.1	Introdução	34
3.2	Definição do modelo de documento para especificação de	requisitos
para aplic	ações móveis sensíveis ao contexto	35
3.2.1	Estrutura das máquinas de estado e sequências de teste	37

	3.2.2	Modelo de documento dos casos de teste	.44
	3.3 A	poio ferramental para a geração de casos de teste para Aplicação	ões
mó	veis Sens	síveis ao contexto (<i>MTContext</i>)	.46
	3.3.1	Descrição da Ferramenta MTContext	.46
	3.3.2	Funcionalidades da MTContext	.47
	3.3.3	Funcionamento da MTContext	.48
	3.3.3.1	Cadastro de Uma Aplicação Móvel	.49
	3.3.3.2	Cadastro de um Dispositivo Móvel	.49
	3.3.3.3	Cadastro de um Elemento	.50
	3.3.3.4	Geração de um teste	.51
	3.3.3.5	Controle dos Testes	.55
CAPÍ	TULO 4 -	ESTUDO DE VIABILIDADE PARA AVALIAÇÃO DA FERRAMEN	TA
MTC	ONTEXT.		.56
	4.1. In	trodução	.56
	4.2. D	efinição do Estudo de Viabilidade	.57
	4.2.1.	Propósito	.57
	4.2.2.	Perspectiva	.57
	4.2.3.	Objetivos específicos	.57
	4.2.4.	Questões e Métricas	.57
	4.3. P	lanejamento do Estudo	.59
	4.3.1.	Formulação de Hipóteses	.59
	4.3.2.	Seleção dos participantes	.64
	4.3.3.	Seleção de Grupos	.65
	4.3.4.	Aplicações a serem testadas	.65
	4.4. A	meaças à Validade do Estudo	.65
	4.4.1.	Validade Interna	.66
	4.4.2.	Validade Externa	.66
	4.4.3.	Validade de Construção	.66
	4.4.4.	Validade de Conclusão	.67
	4.5. E	xecução do Estudo	.67
	4.6. R	esultados do Estudo de Viabilidade	.69
	4.6.1.	Análise Quantitativa	.69
	1611	Casos de teste	73

4.6.1.2. Percentual de Casos de Testes válidos	73
4.6.1.3. Percentual de Casos de Teste Inválidos	74
4.6.1.4. Percentual de Contextos Cobertos	74
4.6.1.5. Percentual de Variações de Contexto	75
4.6.1.6. Tempo	75
4.6.1.7. Testes Válidos Divididos pelo Tempo	75
4.6.1.8. Contextos Cobertos Divididos pelo Tempo	76
4.6.1.9. Variações de Contextos Divididas pelo Tempo	76
4.6.2. Análise Qualitativa	76
4.7. Considerações Finais	81
CAPÍTULO 5 – CONCLUSÃO	83
5.1. Considerações Finais	83
5.2 Contribuições	83
5.3 Limitações	84
5.4 Trabalhos futuros	84
REFERÊNCIAS BIBLIOGRÁFICAS	85
APÊNDICES.	90
Apêndice A Termo de Consentimento livre e Esclarecido – TCLE	90
Apêndice B - Formulário de Caracterização	90
Apêndice C - Requisitos das Aplicações - APP 2 - iFOOD TRUCK	93
Apêndice D - Requisitos das Aplicações - APP 1 - MINHA BICICLETA	94
Apêndice E - Roteiro do Participante – 1° Sessão APP1 – Minha Bicicleta	95
Apêndice F - Roteiro do Participante – 1° Sessão APP2 – iFOODTRUCK	97
Apêndice G - Documento para a especificação dos casos de teste AD HOC	99
Apêndice H - Roteiro do Participante – 2° Sessão	100
Apêndice I - Avaliação Pós Experimento - Teste para AMSC	101

LISTA DE SIGLAS E ABREVIATURAS

APP - (Mobile) Application

AMSC – Aplicações Móveis Sensíveis ao Contexto

CT – Casos de Teste

CTv - Casos de Teste Válidos

CTiv - Casos de Teste Inválidos

V – Variações de Contextos

ExperTS - Experimentação e Teste de Software

ME – Máquina de Estado

MTContext - Context Aware MobileTest

MS – Mapeamento Sistemático

UFAM – Universidade Federal do Amazonas

ÍNDICE DE FIGURAS

Figura 1 Metodologia de Pesquisa adotada	21
Figura 2 Aplicação Tradicional (a) Aplicação Sensível ao Contexto (b)	30
Figura 3 Identificação dos requisitos de uma AMSC.	38
Figura 4 Geração de máquinas de estados e contextos	38
Figura 5 Fórmula de cálculo do total de casos de teste	40
Figura 6 Representação da Geração de Contextos	40
Figura 7 Definição dos casos de teste.	41
Figura 8 Executar os Casos de teste.	41
Figura 9 Visão Geral da Abordagem MTContext	42
Figura 10 Máquina de estados gerada pela abordagem	43
Figura 11 Tela de Cadastro de Aplicação Móvel.	49
Figura 12 Tela de Cadastro de Dispositivo Móvel.	50
Figura 13 Tela de cadastro de um elemento e suas plataformas (a)	51
Figura 14 Tela de cadastro dos dispositivos associados a um elemento (b).	.52
Figura 15 Tela de geração de um teste.	52
Figura 16 Escolha dos Elementos de Contexto.	53
Figura 17 Tela de geração de um teste – Nova Instância	54
Figura 18 Inserção de Comportamento Esperado	55
Figura 19 Tela de geração de um teste - Gerência do teste	55
Figura 20 Estratégia de Execução do Estudo	68

ÍNDICE DE TABELAS

Tabela 1 Lista das Características de Ubiquidade	.26
Tabela 2 Casos de Teste Gerados pela Abordagem	.43
Tabela 3 Inserção dos Comportamentos Esperados	.44
Tabela 4 Resultado Individual de Cada Participante	.71
Tabela 5 Resultados do Teste Estatístico	.72
Tabela 6 Resultados da Análise Qualitativa	.77

ÍNDICE DE QUADROS

Quadro 1 Documento para Especificação de Requisitos	.36
Quadro 2 Exemplo da Abordagem, Inserção de Elementos de Contexto	.42
Quadro 3 Modelo com Informações para Cada Rodada de Teste	.45
Quadro 4 Modelo de Caso de Teste	.45

CAPÍTULO 1 - INTRODUÇÃO

Esta dissertação apresenta uma abordagem para apoio à geração de casos de testes para aplicações móveis sensíveis ao contexto. Nesta abordagem está inserida a proposta de um método de geração de contextos, geração de casos de testes e uma ferramenta de apoio. Neste capítulo serão apresentados a contextualização, a motivação e o problema que será tratado neste trabalho. Também serão apresentados a hipótese, os objetivos, a metodologia de pesquisa adotada, além da organização deste texto.

1.1. Contextualização e Motivação

O cenário de aplicações para dispositivos móveis (que neste trabalho serão chamadas simplesmente de aplicações móveis) só vem evoluindo em tecnologia e se disseminando na população desde a sua criação até os dias atuais. Segundo o IBGE/PNAD (2013), no Brasil, o percentual de pessoas de 10 anos ou mais de idade que tinham celular para uso pessoal passou de 36,6% (55,7 milhões) em 2005 para 69,1% (115,4 milhões) em 2011. Nesses 6 anos, o contingente de detentores de celular para uso pessoal aumentou em 59,7 milhões de pessoas. Ou seja, é uma área que requer bastante atenção.

Segundo Kirubakaran (2013), de acordo com as últimas pesquisas do *Wireless Smartphone Strategies* (WSS), o número de smartphones em uso no mundo ultrapassou a marca de 1 bilhão de unidades pela primeira vez no terceiro trimestre de 2012. Passaram-se 16 anos para a indústria de smartphones chegar a este marco histórico. Olhando outros dados, o *International Data Corporation* (IDC) previu que 182,7 bilhões de aplicativos móveis seriam baixados em 2015. É um aumento de 1.600% em relação aos 10,7 bilhões de aplicativos baixados em 2010.

Enquanto as aplicações móveis foram desenvolvidas inicialmente em sua maioria para o setor de entretenimento, atualmente elas já estão sendo utilizadas em domínios mais críticos, tais como: varejo (ex: locação inteligente de comércio móvel), mídia (revistas e jornais 100% digitais), viagem (reservas de imóveis, check-ins em viagens aéreas, mapas, ofertas, etc.), educação (uso de tablets e aplicativos em salas de aula), saúde (registros de consulta a pacientes, notas de médicos, etc.), finanças (aplicativos para negociação em tempo real, análise de portfólio, movimentação financeira) e sociais (jogos e plataformas de mídia social) (KIRUBAKARAN, 2013).

Com este crescimento e inserção das aplicações móveis na sociedade e o nível de importância que algumas apps exercem, é preciso ter atenção quanto a qualidade dessas aplicações. Barbosa (2000) descreve que atividades agregadas sob o nome de Garantia de Qualidade de Software têm sido introduzidas ao longo de todo o processo de desenvolvimento. Dentre elas podemos citar as atividades de VV&T – Verificação, Validação e Teste, que têm como o objetivo de minimizar a ocorrência de erros e riscos associados a qualidade do software. Dentre as técnicas de verificação e validação, a atividade de teste é uma das mais utilizadas, constituindo-se em um dos elementos para fornecer evidências da confiabilidade do software em complemento a outras atividades. Zhang e Adipat (2005) destacam uma série de questões que foram introduzidas com o advento de dispositivos móveis que precisam ser consideradas durante o desenvolvimento destas aplicações e podem ser avaliadas por meio de testes:

- Conectividade: a conectividade é muitas vezes lenta e incerta em dispositivos móveis. Isto terá um impacto no desempenho de aplicações móveis que utilizam estes recursos.
- Pequeno Tamanho da tela: a fim de proporcionar portabilidade, dispositivos móveis contêm tamanho de tela muito limitado e, assim, a quantidade de informação que pode ser exibida é limitada.
- Resolução Diferente em Displays: a resolução dos dispositivos móveis é reduzida em relação à área de trabalho dos computadores, resultando em imagens de baixa qualidade.
- Capacidade de processamento e poder limitados: a fim de fornecer portabilidade, dispositivos móveis muitas vezes contêm menos capacidade de processamento e poder. Isto limitará os tipos de aplicações que são adequados para dispositivos móveis.
- Diferentes Métodos de acesso: os métodos de entrada disponíveis para dispositivos móveis são diferentes daqueles para computadores desktop e exigem um certo nível de proficiência. Este problema aumenta a probabilidade de entrada errada e diminui a taxa de entrada de dados.
- Contexto celular: quando o uso de aplicativos móveis do usuário não está vinculado a uma única situação. Eles podem também interagir com as pessoas próximas, objetos e elementos ambientais que podem distrair sua atenção.

Junto a esta evolução na plataforma móvel, é preciso ter atenção aos avanços em todas as áreas tecnológicas de dispositivos móveis. Dentre as questões citadas acima,

podemos destacar a questão do contexto de uso do celular, no qual informações do ambiente passam a ser muito importantes. As apps possuem estas características podem ser conhecidas como sensíveis ao contexto, onde novos desafios emergem tanto no desenvolvimento dessas aplicações quanto na avaliação da qualidade, e isto inclui atividades de teste. Wang (2007) explica que isso ocorre porque apps sensíveis ao contexto também apresentam uma nova forma de entrada de dados que atinge seu comportamento em qualquer parte durante qualquer execução.

Esse novo espaço de entrada diz respeito às informações de contexto, que se referem a qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade (DEY, 2001). Por exemplo, no caso de uma aplicação de serviço de táxi sensível ao contexto, a localização do táxi pode ser a informação de contexto utilizada pela aplicação para fornecer as informações adequadas a posição corrente do pedestre. Dessa forma, Wang (2007) enfatiza que as aplicações sensíveis ao contexto são aquelas que adaptam o seu comportamento com base em informações de contexto.

Por tudo que foi mencionado anteriormente, pode-se concluir que, de uma maneira geral, testar uma aplicação móvel sensível ao contexto envolve lidar ao mesmo tempo com os desafios inerentes aos testes de uma aplicação móvel, que precisam, por exemplo, de atenção quanto a mobilidade e entradas de dados, e aos testes de aplicações sensíveis ao contexto, que devem, entre outras coisas, levar em consideração o contexto da aplicação.

1.2. Descrição do Problema

Segundo Harrison (2013), a variedade e a disponibilidade de apps móveis estão se expandindo rapidamente. Os desenvolvedores estão aumentando os serviços que são fornecidos devido ao aumento do poder de processamento disponível nos dispositivos. No entanto, o pequeno tamanho desses dispositivos continua a limitar as maneiras pelas quais os usuários podem interagir com eles.

É preciso garantir a qualidade dessas aplicações e uma das formas de avaliar a qualidade é por meio de atividades de teste de software. Com isso, percebe-se que questões como o pequeno tamanho da tela, conectividade e entradas limitadas têm efeito sobre a usabilidade das aplicações.

Dentre as evoluções no cenário de apps móveis, pode-se destacar o conceito de sensibilidade ao contexto que é uma das características de ubiquidade descritas por Spínola (2010).

Uma aplicação móvel sensível ao contexto possui tanto os desafios de uma app móvel comum quanto os de uma app sensível ao contexto. O seu principal desafio é capacidade de receber diversas informações em qualquer momento de execução que podem fazer com que a app altere seu comportamento e dependendo da informação, diversos tipos de serviços também podem ser oferecidos ao usuário. Com isso, o número de possibilidades de contextos pode ser imenso em apenas uma app.

Esse número de contextos elevado pode tornar a atividade de teste muito dispendiosa, pois será necessário cobrir cada contexto por meio de casos de teste e dependendo da app, esse número pode crescer de forma exponencial. A partir disto, foi feito um levantamento bibliográfico buscando por abordagens que gerem casos de teste para aplicações móveis sensíveis ao contexto (AMSC) e percebeu-se que existem diversas abordagens que tem como objetivo avaliar essas apps, porém o modo de avaliação requer uma grande experiência por parte do testador, pois necessitam identificar possíveis contextos para a app. Algumas abordagens propõem ferramentas de apoio ao teste de aplicações móveis sensíveis ao contexto, porém muitas delas demandam muito tempo para as atividades de identificação das informações a serem inseridas na ferramenta e a inserção dessas informações na ferramenta. Além disso, o número de casos de teste gerados pelas ferramentas é muito extenso. Isto ocorre pelo fato de que o número de possibilidades de variações de contextos de uma aplicação pode ser realmente extenso.

Portanto, a principal motivação para este trabalho é a ausência de métodos formais ou abordagens bem definidas que apoiem a geração de testes a partir de requisitos em AMSC utilizando elementos de contexto e alterações no comportamento da aplicação de forma que esta alteração seja visível na tela do dispositivo. Além disso, esses testes devem utilizar o menor tempo possível e gerar menos testes com uma cobertura igual ou maior ao de abordagens existentes.

1.3. Hipótese

Baseado no contexto e problema apresentados nas seções anteriores, a hipótese definida para este trabalho considera o seguinte cenário:

É possível definir uma abordagem que contribui para a geração de casos de teste com uma maior cobertura de contextos para aplicações móveis sensíveis ao contexto a partir de máquinas de estado em relação a abordagem Ad Hoc baseada no conhecimento do testador.

1.4. Objetivos

Nesta seção serão apresentados os objetivos desta pesquisa.

1.4.1. Objetivo geral

Desenvolver uma abordagem de teste visando a geração de casos de teste em aplicações móveis sensíveis a contexto que possibilite uma maior cobertura dos diferentes contextos associados à aplicação em relação a abordagens tradicionais.

1.4.2. Objetivos específicos

- Fornecer um corpo de conhecimento sobre trabalhos que descrevem teste para aplicações desenvolvidas para a plataformas móveis sensíveis ao contexto;
- Prover uma abordagem de teste que apoie a geração de casos de teste para a cobertura de variações de contextos em aplicações móveis sensíveis ao contexto;
- Desenvolver uma ferramenta que implemente a abordagem proposta;
- Realizar experimentos que possibilitem a avaliação da abordagem proposta em ambientes controlados.

1.5. Metodologia da Pesquisa

A metodologia de pesquisa adotada neste trabalho é baseada na metodologia proposta em Spínola *et al.* (2008), a qual é constituída das seguintes fases: concepção e avaliação da tecnologia, conforme a Figura 1.

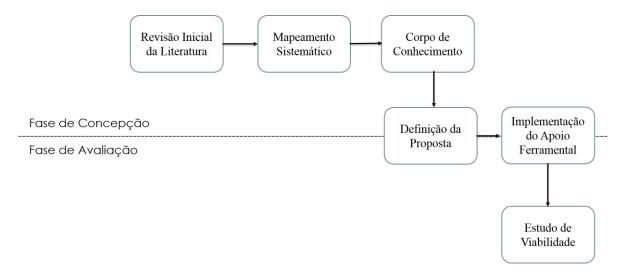


Figura 1 Metodologia de Pesquisa adotada.

1.5.1. Fase de Concepção

A fase de Concepção foi baseada em (SPÍNOLA *et al.*, 2008), que apresenta um conjunto de atividades que utiliza estudos secundários e primários para apoiar a concepção de novas tecnologias de software. Essa fase foi conduzida conforme apresentado na Figura 1. A seguir são descritos detalhes da execução de cada atividade:

- Revisão inicial da literatura: esta atividade foi realizada no período de Março a Outubro de 2013, na qual foi conduzida uma revisão da literatura sobre Avaliação de Usabilidade, Aplicações Móveis e Aplicações Móveis Ubíquas. Esta pesquisa incluiu a busca de aspectos relevantes para a caracterização de avaliações de usabilidade em aplicações móveis sensíveis ao contexto. Os resultados dessa revisão serviram de apoio à próxima etapa da metodologia.
- Levantamento Bibliográfico: esta atividade foi realizada no período de Julho de 2014 a Dezembro de 2014 com o propósito de se obter conhecimento sobre as abordagens de avaliações de aplicações móveis e aplicações móveis sensíveis ao contexto. O planejamento e os resultados desse levantamento serão apresentados no CAPÍTULO 2 - REFERENCIAL TEÓRICO.
- Definição da Proposta: realizada de Janeiro a Março de 2015 a partir do corpo de conhecimento resultante do Mapeamento Sistemático e levantamento bibliográfico, passou-se a trabalhar na abordagem proposta. Após identificação da necessidade de abordagens, métodos e processos para testes de aplicações móveis sensíveis ao contexto, decidiu-se desenvolver uma abordagem de apoio a geração de teste para aplicações móveis sensíveis ao contexto. Os resultados desta etapa serão apresentados no Capítulo 3 deste trabalho.
- Desenvolvimento da Ferramenta de apoio: realizada de Abril de 2015 a Novembro de 2015 após a definição da abordagem, passou-se a desenvolver uma ferramenta, chamada MTContext, que implementasse a abordagem definida. Os resultados desta etapa serão apresentados no Capítulo 3 deste trabalho.

1.5.2. Fase de Avaliação

A fase de Avaliação da tecnologia foi baseada em SHULL *et al.* (2001), cujo propósito é avaliar a tecnologia proposta desde sua definição até sua transferência para a indústria. Essa fase é composta por quatro etapas, das quais até o presente momento foi executada a primeira etapa.

• Estudo de Viabilidade: este estudo foi executado de Dezembro de 2015 a Janeiro de 2016. Uma vez desenvolvida a ferramenta *MTContext*, que implementa a abordagem proposta, o próximo passo foi avaliá-la. Para isso, fez-se necessário planejar e executar um estudo experimental com o objetivo de caracterizar a eficiência e eficácia da abordagem proposta em relação à sua capacidade de gerar um conjunto de casos de teste que possibilite revelar defeitos e com alta cobertura dos contextos presentes em aplicações móveis sensíveis ao contexto. Os resultados deste estudo serão apresentados no Capítulo 4 deste trabalho.

1.6. Estrutura do Documento

Este trabalho está organizado como segue: No Capítulo 2, será apresentado o referencial teórico usado como base para esta dissertação. No Capítulo 3, será apresentada uma abordagem desenvolvida nesta pesquisa para auxiliar a geração de casos de teste para aplicações móveis sensíveis ao contexto e consequentemente facilitar uma possível automatização da execução de testes para essas aplicações, que é a contribuição principal desta dissertação. No Capítulo 4, serão apresentados o planejamento, execução e resultados do Estudo de Viabilidade realizado por meio de um experimento controlado com participantes com perfil de testadores de aplicações móveis. No Capítulo 5 serão apresentadas as conclusões deste trabalho e trabalhos futuros que darão continuidade a esta pesquisa.

CAPÍTULO 2 - REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os conceitos relacionados ao tema desta pesquisa, tais como Aplicações móveis, Aplicações Ubíquas, Teste de Software e trabalhos relacionados. Também serão descritos o planejamento, execução e resultados do mapeamento sistemático realizado sobre teste no contexto de aplicações móveis ubíquas.

2.1. Aplicações móveis

Nesta pesquisa, uma aplicação móvel será abordada de acordo com a definição de Chen et al (2004), que diz que uma aplicação móvel é um software projetado para executar em um smartphone, computadores tablets e outros dispositivos móveis e/ou considerando informações contextuais de entrada. Quatro restrições que podem diferenciar a computação móvel podem ser categorizadas, de acordo com Satyanarayanan (1996): recursos limitados, segurança e vulnerabilidade, variabilidade de desempenho e confiabilidade e fonte de energia finita. Além dessa categorização, Masi et al. (2012) classificam as aplicações móveis em três categorias:

- Aplicações nativas: são aquelas desenvolvidas especificamente para executar em uma plataforma de dispositivo específico. Ela é capaz de acessar os recursos internos do dispositivo, como (por exemplo: câmera, GPS, acelerômetro, gestos, etc.). Normalmente, as aplicações desta categoria devem ser baixadas por meio de lojas de aplicativos (como a *Google Play*, do Android, e App Store, da Apple).
- Aplicações Web: são aquelas aplicações que permitem funções de processamento de informações para ser iniciadas remotamente no servidor web.
 Normalmente, esta categoria de aplicativos são executados por meio de um navegador embutido no dispositivo e não tem acesso a recursos internos do dispositivo.
- Aplicações híbridas (HTML5 e widgets apps): como os aplicativos nativos, os aplicativos híbridos também devem ser baixados a partir de lojas de aplicativos. Eles também ficam armazenados na tela principal do aparelho e pode tirar proveito de todos os recursos internos dos dispositivos. Por outro lado, como aplicações web, que podem ser baseadas em HTML5 e exibidas por meio de um navegador, tendo a totalidade ou parte do conteúdo que foi baixado da web. Aplicativos híbridos permitem o desenvolvimento multiplataforma, usando o

mesmo código HTML para diferentes sistemas operacionais. Eles ainda permitem a compilação, em formato nativo, reduzindo assim os custos de produção.

As aplicações móveis foram desenvolvidas inicialmente em sua maioria para o setor de entretenimento, mas atualmente já estão sendo aplicadas em domínios mais críticos, tais como: varejo (ex: locação-inteligente de comércio móvel), mídia (revistas e jornais 100% digitais), viagem (reservas de imóveis, check-ins em viagens aéreas, mapas, ofertas, etc.), educação (uso de tablets e aplicativos em salas de aula), saúde (registros de consulta a pacientes, notas de médicos, etc.), finanças (aplicativos para negociação em tempo real, análise de portfólio, movimentação financeira) e sociais (jogos e plataformas de mídia social) (KIRUBAKARAN, 2013)

Essas informações apresentadas só denotam a importância das aplicações móveis e a necessidade de estudos para estas plataformas.

2.2. Ubiquidade

A principal definição de ubiquidade ainda é associada ao considerado "pai" da ubiquidade, Mark Weiser (1991), que conceitua como o uso do computador por meio de sua disponibilidade no meio físico, tornando-o efetivamente invisível para o usuário. Ou seja, o computador está de tal forma integrado ao ambiente que sua utilização se dá de forma não intrusiva.

Outra definição é a de que computação ubíqua representa o conceito de computação em todo lugar, fazendo com que o uso da computação e a comunicação sejam transparentes para o usuário (YAU et al., 2002).

Computação Ubíqua pode até mesmo ser confundida com computação móvel. Há relatos de trabalhos que consideram iguais ubiquidade e mobilidade. Porém, outros trabalhos, dentre eles Araújo (2003), explicitam que a computação móvel representa a capacidade do usuário se deslocar com um dispositivo computacional executando um determinado serviço e ainda mantendo conexão, proporcionando a locomoção dos serviços entre ambientes e computação ubíqua une o melhor da computação móvel com o melhor da computação pervasiva que é capacidade de os dispositivos serem embutidos no ambiente físico de forma não intrusiva.

A definição que será adotada neste trabalho foi obtida por meio de uma revisão sistemática realizada por Spínola et al. (2007): "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".

Além dessa definição, Spínola et al. (2012) definiram algumas características funcionais presentes na computação ubíqua que podem ser vistas na Tabela 1.

Tabela 1 Lista das Características de Ubiquidade.

Característica	Descrição
Onipresença de serviços	Permitir a movimentação física do usuário, dando a ele a percepção de estar levando consigo os serviços computacionais.
Invisibilidade	Capacidade de estar presente nos objetos de uso do dia-a-dia, descaracterizando, do ponto de vista do usuário, a "utilização" de um computador e acentuando a percepção de objetos ou dispositivos que proveem serviços ou algum tipo de "inteligência". Com isto, procura-se permitir alternativas apropriadas para as interfaces gráficas tradicionais utilizadas nas soluções desktop, de forma a privilegiar formas mais naturais de entrada de dados tais como reconhecimento de escrita, fala, gestos, expressões faciais, ou movimentos ou ainda integrar essas formas de tal modo que a interface seja minimamente percebida pelo usuário.
Sensibilidade ao contexto	Capacidade de coletar informações sobre o ambiente onde está sendo utilizado.
Comportamento adaptável	Capacidade de, dinamicamente, adaptar os serviços disponíveis ao ambiente onde está sendo utilizado dentro de suas limitações.
Captura de Experiências	Capacidade de capturar e registrar experiências para uso posterior.
Descoberta de serviços	Construir serviços proativamente de acordo com o ambiente que se encontra. A aplicação deve interagir com o ambiente e permitir que o usuário também o faça a fim de descobrir novos serviços ou informações para atingir o objetivo desejado.
Composição de funcionalidade	Capacidade de, a partir de serviços básicos, montar uma determinada funcionalidade requerida pelo usuário.
Interoperabilidad e espontânea	Capacidade de interagir com outros dispositivos durante a sua operação conforme a sua movimentação.
Heterogeneidade de Dispositivos	Prover mobilidade de aplicação através de dispositivos heterogêneos.
Tolerância a falhas	Capacidade de se adaptar diante de falhas no ambiente (por exemplo, disponibilidade on-line/off-line).

Sendo assim, este trabalho está embasado nas definições de Spínola *et al.* (2012) sobre ubiquidade e suas características que foram apresentadas acima.

2.3. Aplicações Sensíveis ao Contexto

Em um estudo de Abowd et. al. (1999), é visto que em computação sensível ao contexto, uma aplicação é sensível ao ambiente computacional na qual é executada e ela se

adapta ou reage de acordo com o seu contexto computacional, de usuário, físico ou de tempo.

O contexto, por sua vez, pode ser definido como sendo qualquer informação que possa ser utilizada para caracterizar a situação de uma entidade, sendo que as entidades podem ser definidas em fatores humanos (ex: usuário, ambiente social e tarefas) e ambientes físicos (ex: localização, infraestrutura e condições físicas) (DEY, 2001).

Sendo assim, a diferença entre aplicações sensíveis ao contexto e as aplicações tradicionais é que as sensíveis ao contexto utilizam o contexto para configurar o seu comportamento. Segundo Vieira et. al. (2009), uma aplicação sensível ao contexto pode considerar tanto as informações explícitas fornecidas pelos usuários, as inferidas por meio de raciocínio e também aquelas percebidas a partir do monitoramento do ambiente. Todas essas informações obtidas de forma não-explícita são chamadas de informações contextuais. Já uma aplicação tradicional, não sensível ao contexto, possui um processo de informação baseado apenas nas solicitações e entradas explícitas do usuário.

2.4. Teste de Software

Teste de Software consiste em uma investigação experimental conduzida para prover informações aos usuários e envolvidos no processo sobre a qualidade do software em teste, no contexto no qual este será operado. Isso inclui, mas não consiste apenas em, um processo de executar um programa com a intenção de revelar falhas, apresentando uma análise dinâmica do produto, com atividades planejadas previamente e de modo sistemático (KANER, 2006). O teste de software também pode ser considerado como o processo de executar um programa ou sistema com a intenção de revelar falhas (MYERS, 1979).

Delamaro et al. (2007) salientam que a atividade de teste é uma atividade complexa e que erros podem ser ocasionados por diversos fatores. Além disso, a importância do teste de software e sua relação com a qualidade devem ser enfatizadas. PRESSMAN (2006) diz que toda e qualquer estratégia de teste deve incorporar planejamento, projeto de casos de teste, execução e avaliação de dados resultantes. Esses testes podem ser realizados por meio de casos de teste. Um caso de teste é uma especificação de como o software deve ser testado através da definição das entradas e saídas esperadas, bem como das condições sob as quais o teste deve ocorrer (PRESSMAN, 2006).

Contudo, Brito (2012) ressalta que é indispensável no processo de desenvolvimento de software a utilização de métodos, técnicas e ferramentas que permitam a realização da atividade de teste de maneira que auxiliem no aumento da produtividade, da qualidade e da diminuição de custos desta atividade. Pois, segundo Pressman (2006), os custos

associados à atividade de teste, podem chegar a mais de 50% do custo total de um projeto. Além disso, ferramentas e abordagens que possam auxiliar a atividade de teste podem apoiar na redução do custo da atividade de teste.

2.4.1. Teste de Aplicações móveis

No desenvolvimento de um *app*, o teste é uma fase obrigatória que tem como objetivo certificar que os dispositivos móveis são confiáveis, robustos e "livres de erros" (DANTAS 2009). Para *apps*, essa tarefa na prática não é trivial e requer adaptações no processo de teste tradicional, tornando-o mais complexo. Essa complexidade envolve tanto questões do ambiente desktop e ainda se adicionam as limitações do ambiente móvel que são incorporadas no processo de teste para aplicações móveis. (LIONBRIDGE 2006)

Como por exemplo, Dantas (2009) aborda, em forma de questões, problemas típicos enfrentados por testadores durante os testes no ambiente móvel:

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

Teste para aplicações móveis significa diferentes tipos de teste a serem aplicados para os diferentes tipos de aplicações móveis (aplicações nativas, web e híbridas) que executam sob determinada plataforma móvel. Alguns requisitos e especificações diferenciam a engenharia de aplicações móveis de outras plataformas, e isso possui influência na forma como testes devem ser realizados. Esses testes devem usar métodos e ferramentas de teste de software bem definidos a fim de garantir a qualidade em funções, comportamentos, desempenho e qualidade de serviço, assim como atributos como mobilidade, usabilidade, interoperabilidade, conectividade, segurança e privacidade (GAO, 2014).

Os testes nas aplicações móveis podem ser feitos por meio de emuladores que simulam a aplicação no próprio computador. Porém, segundo Zeidler et al. (2007), o uso de emuladores não demonstrou ser um meio de substituição dos testes nos próprios dispositivos, portanto a tarefa de testar no próprio dispositivo ficou como trabalho para a equipe de testes. Além disso, para AMSC ainda se faz necessária a utilização de devices

devido as inúmeras situações de contextos que surgem provenientes da interação com sensores nativos dos dispositivos móveis que não podem ser identificadas utilizando apenas os emuladores.

2.4.2. Teste de Aplicações Móveis Sensíveis ao Contexto

Diferentemente de aplicações tradicionais, onde o comportamento está incluso no código implementado, aplicações móveis sensíveis ao contexto têm parte do seu comportamento determinado de acordo com o contexto (TSE et al., 2004). Sendo assim, Wang et al. (2007) esclarecem que o problema em se testar aplicações sensíveis ao contexto está relacionado ao desafio de prever todas as mudanças de contexto relevantes e quando elas podem impactar o comportamento deste tipo de aplicação. A validação de aplicações móveis sensíveis ao contexto está ligada principalmente à emulação física do dispositivo móvel, a implantação física e lógica a um subconjunto acessível dos cenários de usuários e a adaptação das metodologias de geração de casos de teste existentes (SATOH, 2003) (TSE; YAU; CHAN; LU; CHEN, 2004).

Segundo Lu (2009) existem quatro desafios em se testar aplicações sensíveis ao contexto:

- Formatos ricos. Dados de contexto podem ser capturados a partir de uma variedade de diferentes fontes dinamicamente. Os dados obtidos são, então, representados em formatos ricos e hierarquias semânticas. Testadores precisam cuidadosamente replicar ou simular valores de contexto enquanto projetam entradas de teste. Além disso, Lu (2009) destaca que ao desenvolver uma ferramenta ou método de teste deve-se lembrar que contextos têm estruturas de representação interna que variam de simples modelo chave-valor para representações gráficas e ontologias;
- Contextos Voláteis. TSE et al. (2004) e WANG et al. (2007) concordam que contextos podem variar rapidamente. Quando a execução de um programa usa um contexto particular para derivar sua resposta ao usuário, o contexto já pode ter sido substituído por um novo contexto. A execução de um programa é então fragmentada em vários segmentos, tais que diferentes segmentos podem se referir a diferentes contextos. Assim, diferentemente de um programa tradicional, no qual é produzido o mesmo resultado para o mesmo valor de entrada, a entrada de uma aplicação sensível ao contexto pode induzir diferentes execuções resultantes de acordo com os diferentes contextos sob os quais a aplicação executou;

- Sensibilidade ao contexto egocêntrica. Segundo Lu (2009), entidades computacionais são usualmente modeladas como serviços e tem comportamento egocêntrico se preocupando apenas com as informações de contexto que lhe interessam. Dessa forma, o contexto observado, por exemplo, na presença de ruído pode não ser exatamente o mesmo da situação esperada para disparar um dado serviço fazendo com que o serviço não seja invocado quando deveria;
- Soft Computing. Contextos físicos são imprecisos por natureza. Logo, quando a execução de um programa gera um resultado, é difícil determinar quando ele é (1) correto; (2) errado devido a falha de software; ou (3) errado devido à imprecisão da informação de contexto.

A evidência da diferença entre teste para aplicações não sensíveis ao contexto para teste para aplicações sensíveis ao contexto pode ser vista na Figura 2. Fonte: (VIEIRA; TEDESCO; SALGADO,2009)

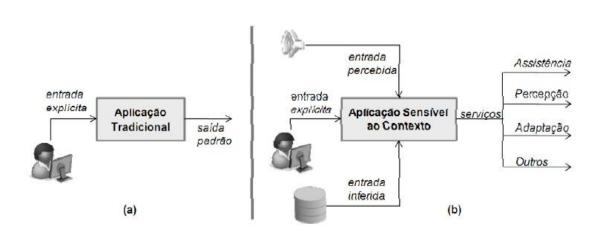


Figura 2 Aplicação Tradicional (a) Aplicação Sensível ao Contexto (b).

É possível perceber que na situação (a) existe uma entrada padrão e consequentemente uma saída padrão. Diferentemente da situação (b), em que existem tantas possíveis entradas e igualmente proporcional tantas possíveis saídas que o teste para essas aplicações requer mais cuidado e atenção, pois o número de possibilidades de saída só aumenta de acordo com o contexto em que o usuário se encontrar e as informações recolhidas como entrada para aplicação.

2.5. Trabalhos relacionados

Foi realizada uma investigação na literatura em busca de pesquisas que envolvessem teste para aplicações móveis sensíveis ao contexto. É importante mencionar

que esses tópicos foram definidos porque representam aspectos fundamentais dessa pesquisa. Para o levantamento de trabalhos relacionados, foram obedecidos alguns princípios do processo de mapeamento sistemático, tais como a definição de strings de busca e locais de pesquisa. Como locais de buscas, utilizaram-se as bases Scopus¹ e o IEEE Explorer², bem como os anais do Simpósio Brasileiro de Engenharia de Software (*SBES*); do Simpósio Brasileiro de Qualidade de Software (*SBQS*); do Brazilian Workshop on Systematic and Automated Software Testing (SAST); e do Simpósio Brasileiro de Computação Ubíqua e Pervasiva (SBCUP). Além disso, buscando por *grey literature* (ex: relatório técnicos e trabalhos em progresso), foram executadas buscas no Google³ e no Google Scholar⁴.

Existem vários trabalhos que abordam diferentes tipos de teste para aplicações móveis sensíveis ao contexto, porém houve dificuldade para encontrar trabalhos que apoiassem a geração de casos de teste para essas aplicações por meio de sua interface.

2.5.1. Geração de Teste para Aplicações Móveis Sensíveis ao Contexto

Yu et al. (2005) apresentam princípios e práticas de testes ágeis que podem servir para aplicações sensíveis ao contexto, uma vez que as aplicações acabam sendo desenvolvidas e implantadas de forma iterativa e evolutiva. Nesse artigo, eles exploram um framework de testes para serviços baseados em localização que oferece testes funcionais e também oferece uma gama completa de testes não-funcionais que estas aplicações requerem.

Cano et. al. (2005) descrevem um teste de desempenho em relação ao uso da tecnologia sem fio Bluetooth para fornecer suporte de rede para aplicações sensíveis ao contexto. Apesar da aplicação ser sensível ao contexto, o teste consistiu em apenas na verificação desta aplicação ubíqua desenvolvida pelos autores. Esta verificação foi feita por meio de um banco de casos de teste já existente para teste e simulação de desempenho de aplicações, ou seja, sem nenhum foco nas características de ubiquidade ou nas peculiaridades de contextos.

Em Lu et. al. (2006), é apresentada uma nova família de critérios de teste para medir a abrangência de seus conjuntos de teste. Ela decorre a partir de informações de fluxo de dados sensíveis ao contexto. Em primeiro lugar, estuda a evolução de contextos, que são

¹ http://www.scopus.com/

² http://ieeexplore.ieee.org/

³ https://www.google.com.br/

⁴ https://scholar.google.com.br/

informações ambientais pertinentes para uma aplicação. Em seguida, ele propõe associações de fluxo de dados sensíveis ao contexto e os critérios de teste e algoritmos correspondentes são gerados.

Raymundo et al. (2012) apresentam uma ferramenta, denominada *CraftContext*, capaz de auxiliar a fase de testes no desenvolvimento de aplicações sensíveis a contexto. *CraftContext* oferece um ambiente 3D de simulação de contexto rico em detalhes e variedade de recursos, cujo acesso ocorre por intermédio de uma API robusta e portável. Sua vantagem em relação às principais ferramentas existentes é a sua adaptabilidade a domínios diversos.

Hesenius et al. (2014) propõem uma abordagem baseada em modelos para melhorar o teste de aplicações móveis sensíveis ao contexto, deduzindo casos de teste a partir de modelos de sistemas de tempo/design. Usando uma versão personalizada de um framework de testes chamado calabash-android.

Wang (2007), verifica casos de testes existentes e melhora-os por meio de três ações. Primeiro identifica o objetivo dentro da aplicação e aponta qual a informação de contexto pode afetar o comportamento da aplicação, depois gera potenciais variantes para cada caso de teste existente que explora a execução de diferentes sequências de contexto, e por último ele tenta dinamicamente executar a aplicação para as sequencias de teste gerado. Parte da estrutura do trabalho de Wang (2007) foi aproveitado como base para o desenvolvimento da ferramenta apresentada no Capítulo 3.

O mais próximo do objetivo dessa dissertação foi apresentado em Amalfitano et al. (2013), onde os autores abordam o problema de testar aplicações móveis como um sistema orientado a eventos, tendo em conta ambos os eventos de contexto e eventos GUI. Ele gera os contextos por meio de padrões de eventos e esses eventos são definidos de acordo com elementos da interface. Porém ele não leva em consideração os elementos nativos dos dispositivos e não confirmam se um contexto foi mesmo executado ou se a tela está de acordo com requisitos pré-determinados.

Finalmente, Samsel *et al.* (2014) descrevem uma abordagem com o intuito de reduzir problemas cognitivos para os usuários. Eles utilizam uma aplicação móvel (protótipo) para o transporte público que foi desenvolvida baseada no princípio da "Teoria da Informação em Cascata" para simplificar o uso e, adicionalmente, para usar um *backend* que permite integrar dados de serviços diferentes e assim unificar as apresentações das aplicações. Porém mesmo após desenvolver uma aplicação que se preocupa com os elementos da interface da aplicação (Graphical User Interface - GUI), esta abordagem não utiliza uma

técnica de teste específica para avaliar a eficácia de suas apresentações de telas para o usuário. Para avaliar a aplicação desenvolvida foi feito apenas um teste de usuário inicial.

Apesar das diversas técnicas expostas nesta seção, nenhuma dessas técnicas apresentam em suas etapas foco em teste por meio dos elementos nativos pertencentes a aplicações móveis sensíveis ao contexto que podem variar de acordo com a plataforma ou dispositivo móvel. Sendo assim, nenhuma destas técnicas foi utilizada como comparação para a abordagem proposta neste trabalho.

2.6. Considerações Finais

Ao longo deste capítulo foram apresentados os principais conceitos necessários ao entendimento e desenvolvimento desta pesquisa, bem como trabalhos relacionados. Estes conceitos são importantes para mostrar a relevância deste trabalho e motivá-lo com os resultados já apresentados na literatura técnica. Baseado nisso, Amalfitano et al. (2013) destacam que um fator de complexidade relevante no desenvolvimento e teste de aplicativos móveis é dada por sua sensibilidade às mudanças no contexto em que eles correm. Como exemplo, aplicativos rodando em um smartphone podem ser influenciados por alterações de local, ligações, por movimentos do dispositivo e muitas outras tipologias de eventos de contexto. Além disso, as aplicações móveis estão cada vez mais baseadas em localização; ou seja, a sua funcionalidade está se tornando tanto interativa e sensível ao contexto. Combinado com um aumento global na complexidade dos dispositivos de entrega de tais serviços, e um aumento do número de possíveis grupos em que eles podem participar, estes sistemas precisam muito de abordagens de teste que apoiem o teste dessas aplicações (YU, 2005).

Por tudo que foi apresentado neste capítulo, é possível observar que embora tenham sido encontrados alguns trabalhos relacionados, nenhum leva em consideração os elementos nativos dos dispositivos móveis que podem variar de sistema operacional e até mesmo variar de tipos de dispositivos. Sendo assim, com o objetivo de melhorar a qualidade dos testes em aplicações móveis sensíveis ao contexto, foi desenvolvida uma abordagem baseada em comportamentos visuais esperados de uma aplicação móvel, que visa apoiar a geração de teste de contextos alcançados na aplicação por meio de seus elementos nativos dos dispositivos. O objetivo com a abordagem é atuar na qualidade de aplicações móveis sensíveis ao contexto gerando casos de teste que cubram os possíveis contextos que a aplicação possa ser direcionada por meio de máquinas de estados que gerarão os contextos e a partir dessas máquinas de estados serão gerados casos de testes para cada máquina de estado gerada. No próximo capítulo será apresentada com detalhes a abordagem criada.

CAPÍTULO 3 – ABORDAGEM DE GERAÇÃO DE TESTE PARA APLICAÇÕES MÓVEIS SENSÍVEIS AO CONTEXTO

Neste capítulo é descrito o processo de criação da abordagem MTContext que consiste em uma abordagem para geração de testes para aplicações móveis sensíveis ao contexto, bem como o processo de desenvolvimento da ferramenta que a implementa.

3.1 Introdução

Um mapeamento sistemático publicado em REIS et al., (2015) sobre avaliações de usabilidade para aplicações móveis ubíquas revelou a existência de poucos trabalhos que avaliam aplicações sensíveis ao contexto. Porém, dentre as características de ubiquidade, sensibilidade ao contexto é a que se apresentou com mais trabalhos abordando o tema. Esse fato contribuiu para o direcionamento desta pesquisa, pois mostrou uma necessidade de mais estudos nesta área. Os desafios no teste de aplicações móveis sensíveis ao contexto vêm sendo discutidos na literatura técnica. As peculiaridades que estas aplicações trazem consigo, tornam os seus testes desafiadores. Sensibilidade ao contexto é um tema emergente no mundo e suas apps estão sendo ainda mais desenvolvidas e publicadas em lojas de apps. Logo, ele precisa de bastante atenção de pesquisadores. Por isso, iniciativas de garantir a qualidade dessas aplicações devem ser tomadas para que defeitos possam ser identificados e corrigidos.

A abordagem MTContext foi criada a partir de padrões existentes na literatura técnica e a experiência do grupo de pesquisa em Experimentação e Teste de Software (ExperTS) da Universidade Federal do Amazonas (UFAM) em projetos de software na indústria. O levantamento bibliográfico realizado para identificar os testes realizados em AMSC serviu como base teórica para a abordagem criada neste trabalho, tanto para a concepção da especificação de requisitos, extraídas de técnicas para aplicações móveis, quanto a geração de teste de AMSC encontradas em várias publicações neste levantamento bibliográfico. Além disso, a geração automatizada de testes foi baseada em máquinas de estados vistas em Wang (2004) e Pinheiro (2014), também encontrados neste levantamento bibliográfico. Essas técnicas levaram à execução das seguintes tarefas:

- Definição o modelo de documento para especificação de requisitos;
- Definição a estrutura de máquina de estado a ser gerada;

- Definição modelo/estrutura do documento do caso de teste; e
- Desenvolvimento uma ferramenta que implemente a abordagem proposta (MTContext);

As atividades realizadas no desenvolvimento desta pesquisa serão detalhadas a seguir.

3.2 Definição do modelo de documento para especificação de requisitos para aplicações móveis sensíveis ao contexto

Para a definição do documento de especificação de requisitos, buscou-se primeiro por documentos já existentes. Foram encontrados alguns artigos que definiam os requisitos de aplicações sensíveis ao contexto, porém alguns desses documentos eram muitos extensos, ou a descrição era muito técnica ou não seguiam nenhum padrão de linguagem. Isto é, dificultaria possíveis gerações de testes mais rápidos e necessitariam um alto entendimento do testador com o projeto e com teste de software para pode entender o documento e assim criar casos de teste. Dentre esses artigos encontrados, são citados dois:

Santos (2013) apresenta um método que utiliza como base especificações textuais de casos de uso com informações sobre: funcionalidade, variabilidade da linha (relacionado a linha de produto de software) e como o contexto afeta os produtos finais. O ambiente também contém o modelo de documento para caso de uso que fornece suporte ao uso do método e a ferramenta de apoio, que permite a modelagem de casos de uso segundo esse modelo de documento e implementa o método proposto. A especificação de requisitos por meio de casos de uso é bem ampla e por isso torna difícil a geração de casos de teste de uma forma fácil, rápida e eficaz. Sendo assim, este método proposto por Santos (2013) não se encaixava na abordagem a ser proposta nesta pesquisa.

Mota (2013) propõe uma abordagem de apoio à especificação de requisitos funcionais de ubiquidade que possibilita tratar questões relacionadas à garantia da qualidade, baseada em um metamodelo elaborado, que organiza as características e fatores de ubiquidade e suas interrelações. Este modelo foi integrado a uma abordagem para especificação de requisitos de aplicações web para tornar a especificação mais abrangente ao permitir a descrição detalhada do comportamento sistêmico, tendo em vista que esse tipo de aplicação possui características comuns que favorecem a integração. Esta abordagem abrange todas as descrições de aplicações ubíquas de forma detalhada. No entanto, a necessidade de se obter um documento com especificações simples e diretas que levem à descrição apenas dos fatores que possam alterar o comportamento da aplicação, também faz desta abordagem não ideal para a ideia de documento a ser proposta nesta dissertação.

Tendo em vista os exemplos de especificações de requisitos citados acima, decidiuse por criar um documento que reunisse as melhores características coletadas na literatura. Para composição do documento, primeiro foi avaliado quais seriam os campos mais importantes a serem colocados no documento de tal forma que se já houver algum outro documento como por exemplo casos de uso, histórias ou cenários, a extração das informações a serem colhidas para o documento não se torne maçante e demande muito tempo. O Quadro 1 mostra os campos e exemplos de uso desses campos como é descrito a seguir:

CamposExemplosElementoConexãoTipo de elementoNominalVariações do elementoLongeelementoDesligar GPS e religar após 20minutosPertoLigar Flash do dispositivo

Quadro 1 Documento para Especificação de Requisitos

- Elemento: este campo refere-se a formas de recolhimentos de dados de entrada. Como mostrado na subseção 2.4.2, sensibilidade ao contexto possui várias opções de entrada de dados e com eles existe a forma de recolhimento dessas entradas, seja ela uma entrada percebida, inferida ou explícita. Aqui chamado de Elemento, pode também ser conhecido como variáveis de entrada. Os elementos identificados nesta dissertação para dispositivos móveis são Conexão, Localização, Rotação, Bateria, Tempo, Temperatura e outros sensores. É possível ter mais de um elemento em um cenário.
- **Tipo de Elemento:** é uma identificação que possibilita um detalhamento sobre o elemento. São eles:
 - o **Intervalar:** quando o elemento vem em forma de intervalos. Exemplo:

'Quando o valor de Localização estiver entre 2 e 4 metros, o dispositivo deve ligar o flash de sua câmera.' O elemento intervalar aqui é "entre 2 e 4 metros".

Nominal: quando o elemento vem apenas descrito sem intervalo definido.
 Exemplo:

Quando o valor de Localização estiver perto do ponto de chegada, o dispositivo deve ligar o flash. O elemento nominal aqui é o valor subjetivo "perto".

 Variações do Elemento: onde devem ser separadas e descritas todas as possíveis variações para o elemento. Exemplo:

Conexão possui três variações. São elas 3G, WIFI e Off.

 Ações para o elemento: este campo deve ser preenchido para cada variação do elemento. Ou seja, cada variação de elemento deve ter uma ação no dispositivo que o acione. Exemplo:

Quando o valor de Localização estiver entre 2 e 4 metros, o dispositivo deve ligar o flash de sua câmera. O disparo para a ação do elemento seria estar entre 2 e 4 metros. A ação seria "Ligar o Flash".

O documento foi criado com a intenção de extrair informações de outros documentos, como por exemplo casos de uso, cenários, dentre outros que seriam adequados a Aplicações Móveis Sensíveis ao Contexto (AMSC). Um bom exemplo de uso do documento é: para cada caso de uso de uma AMSC, deve ser realizada uma extração com o documento. Ou seja, em cada caso de uso, deve ser identificado quais os Elementos, Tipos de Elementos e as Variações de Elementos existentes na aplicação. É importante ressaltar que cada extração realizada será considerada uma rodada de teste a ser criada.

3.2.1 Estrutura das máquinas de estado e sequências de teste

Nesta fase, foi definido como seria o funcionamento do algoritmo que faria a identificação dos Elementos de Contexto e em seguida geraria as sequências de teste ou casos de teste a partir desses Elementos de Contexto.

O objetivo nesta fase é gerar máquinas de estado a partir dos elementos de contexto de uma aplicação móvel sensível ao contexto de forma que casos de testes possam ser gerados a partir dessas máquinas de estado. Assim, as exclusividades de AMSC seriam reproduzidas em máquinas de estado e qualquer outra abordagem tradicional que gerasse casos de testes a partir de ME poderiam ser utilizadas, independente de tecnologia ou plataforma. A forma de geração de casos de teste a partir de máquinas de estado pode ser

encontrada em (WANG, 2004) e (PINHEIRO, 2014). A abordagem proposta foi chamada de *Context Aware Mobile Test* (sigla *MTContext*). Ela se divide em 3 (três partes), sendo:

- 1. Identificação dos requisitos de sensibilidade ao contexto da aplicação móvel;
- 2. Geração de máquinas de estados e contextos a partir das informações obtidas no passo anterior; e
- 3. Definição dos casos de teste a partir dos contextos gerados no passo anterior.

Todo o processo de funcionamento da abordagem a partir da identificação dos elementos de contexto dentro dos requisitos de uma aplicação móvel será descrito a seguir.

A Figura 3 apresenta a primeira parte da *MTContext*. Inicialmente o Testador irá identificar as informações de contexto que geram alguma alteração ou adaptação na aplicação a ser testada. Essas informações estarão presentes nos requisitos da aplicação, seja ela casos de uso, cenários, dentre outros. Após a identificação, essas informações de contexto serão inseridas na abordagem por meio do documento de especificação de requisitos de aplicações sensiveis ao contexto, *reqContext*.

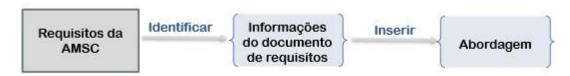


Figura 3 Identificação dos requisitos de uma AMSC.

A Figura 4 apresenta a segunda parte da *MTContext*. Nesta parte, a abordagem irá gerar uma máquina de estado a partir das informações inseridas no campo *Variações de Elementos*, apresentados na subseção 3.2.1.

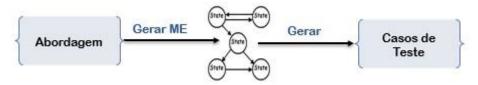


Figura 4 Geração de máquinas de estados e contextos.

Uma máquina de estados finitos - MEF é uma representação de uma máquina composta por estados e eventos, que correspondem a transições entre os estados. Uma transição é caracterizada por dois eventos: um de entrada e um de saída. A máquina pode estar em apenas um estado por vez. Ao ocorrer um evento de entrada, a máquina pode responder com um evento de saída e uma transição para outro estado (podendo ser para o mesmo estado) (PINHEIRO, 2014).

Existem dois tipos de MEFs: *Mealy* e *Moore*. Na máquina de *Mealy*, os eventos de saída estão associados às transições, ou seja, ao ocorrer um evento de entrada, o evento de saída ocorre durante a mesma transição. Na máquina de *Moore*, os eventos de saída estão associados aos estados; sendo assim, o evento de saída ocorre ao final da transição, no seu estado destino (PINHEIRO,2014).

Neste trabalho, a máquina de *Mealy* foi a escolhida por ser mais utilizada para geração de casos de teste. A máquina de estados dentro da *MTContext* é composta da seguinte maneira:

- Estado: é um contexto, um agrupamento de *variações de elementos*.
- Transição: composto por eventos.
 - Evento de entrada uma variação de elemento a ser alterada. Essa variação de elemento é dependente de um elemento, ou seja, uma variação só pode ser alterada/trocada por outra, se esta pertencer ao mesmo elemento.
 - Evento de saída a composição do próximo estado com o elemento alterado incluso com as demais variações de elementos.

Sendo assim, uma máquina de estado nesta abordagem é definida com estados que são compostos por um número finito de variações de elementos inseridas pelo testador de forma que uma transição de um estado para o outro é feita por meio de eventos de entrada e saída, sendo que um evento de entrada é uma alteração em uma determinada variação de elemento e um evento de saída é a nova composição do próximo estado.

Ainda na Figura 4, é apresentada a geração de casos de teste que é feita após a geração de máquinas de estados. Para iniciar a geração de casos de teste desta abordagem, primeiro foi feita uma ordenação no momento de geração da máquina de estado de modo que uma transição de um estado para o outro só ocorrerá se apenas uma variação for alterada por cada estado até que todas as possíveis alterações entre as variações sejam feitas e, assim, a máquina estará completa.

Tendo dito isso, os casos de teste são formados por cada estado/contexto da máquina de estado gerada e o número total de casos de teste. Matematicamente, o total de casos de teste *TC* é igual ao produto cartesiano dos os elementos do conjunto *A* por *B* (Figura 5). Sendo que o total de conjuntos é definido pelo total de elementos nativos utilizados no teste da *app*. Ou seja, se durante o teste forem utilizados 3 elementos nativos (bateria, conexão e localização), a equação será composta por 3 conjuntos e o total de casos de teste *TC* será o produto cartesiano dos elementos *A* por *B* por *C*.

Total de Casos de Teste TC: A x B = $\{(x, y) \mid x \in A \in y \in B\}$.

Figura 5 Fórmula de cálculo do total de casos de teste.

Ou seja, dado um aplicativo que contém 2 elementos nativos do dispositivo móvel (Bateria e Conexão) e cada um desses elementos possuem variações, cada elemento nativo será compreendido como Conjunto e as variações serão os elementos dos conjuntos. Sendo assim, o total de casos de teste *TC* para os conjuntos representados por *A*= (*a*, *b*) e *B*= (*c*, *d*, *e*) será o produto cartesiano de *A* x *B*. O total de casos de teste *TC* criado pela *MTContext*, neste exemplo será 6 e consequentemente o número de contextos cobertos por esses casos de teste também deverá ser 6.

TC: A x B=
$$(a, b)$$
 x (c, d, e)
(A x B) = 2 * 3 = 6

A sequência formada por este produto cartesiano será:

TC:
$$A \times B = (a, b) \times (c, d, e)$$

 $A \times B = \{(a, c), (a, d), (a, e), (b, c), (b, d), (b, e)\}$

Uma outra representação pode ser dada na Figura 6 em que todos os elementos do Conjunto **A** estão ligados a todos elementos do Conjunto **B**.

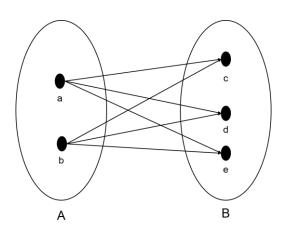


Figura 6 Representação da Geração de Contextos

Finalmente, a Figura 7 apresenta a terceira parte da *MTContext*. Nesta fase o testador irá informar quais são os comportamentos esperados para cada contexto gerado.



Figura 7 Definição dos casos de teste.

O testador deverá inserir, se houver, uma captura da tela (*screenshot*) que ilustre o comportamento esperado desta aplicação. Mais detalhes sobre o porquê de inserir uma *screenshot* serão apresentados na próxima subseção 3.2.3.

A Figura 8 apresenta uma quarta parte da abordagem em que será possível executar os casos de testes gerados pela *MTContext* por meio da ferramenta Sikuli. Chang (2010) descreve a Sikuli como uma ferramenta que utiliza abordagem visual para pesquisar e automatizar testes GUI usando *screenshots*. Os dados de saída da *MTContext* serviriam como dados de entrada da Sikuli que executaria os casos de teste e faria a comparação de resultado esperado com resultado obtido na aplicação por meio de comparação visual entre as *screenshots* obtidas. Esta quarta parte não foi implementada até o presente momento, por isso será melhor discutida na subseção seguinte 3.2.3 e na seção 5.4 de trabalhos futuros.



Figura 8 Executar os Casos de teste.

Com isso, a abordagem MTContext completa ficaria como mostrado na Figura 9.

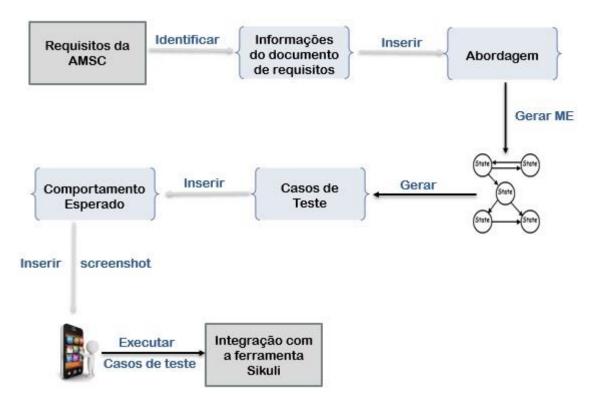


Figura 9 Visão Geral da Abordagem MTContext.

Para melhor compreensão da abordagem e entendimento do funcionamento do algoritmo de geração de máquinas de estado e sequencias de teste, um exemplo será utilizado para ilustrar o funcionamento da mesma:

Um aplicativo chamado iFOODTRUCK é executado (aberto) automaticamente todas as vezes que o dispositivo móvel se conecta à Internet ou ativa a localização (GPS). Por outro lado, o mesmo aplicativo, se estiver aberto, se encerrará (fechará) automaticamente sempre que o dispositivo móvel se desconectar da Internet ou desativar a localização (GPS).

Os seguintes passos deverão ser executados:

1. Identificar as informações de Contexto e Inserir na abordagem por meio do documento de especificação de contextos.

Instanciando o primeiro passo da abordagem, as informações de contextos identificadas no cenário apresentado, ficariam descritas conforme mostrado no Quadro 2.

Quadro 2 Exemplo da Abordagem, Inserção de Elementos de Contexto

Campos	Exemplos	Ação para a variação dos
--------	----------	--------------------------

Elemento 1	Conexão	elementos					
Tipo de elemento	Nominal						
Variações do	ON	Ligar App					
elemento	OFF	Desligar App					
Elemento 2	Localização	Ação para a variação dos					
Elemento 2 Tipo de elemento	Localização Nominal	Ação para a variação dos elementos					
		1 , .					

2. A abordagem irá gerar a máquina de estado e os casos de teste.

A Figura 10 mostra como ficará a máquina de estado gerada para o exemplo acima. Um estado é um agrupamento de variações de elementos $q0 = L:on\ C:on$ e as transições são, *Entrada: Variação de elemento a ser trocada C=off* e Saída: o próximo estado já com a alteração feita $q1=L:on\ C:off$.

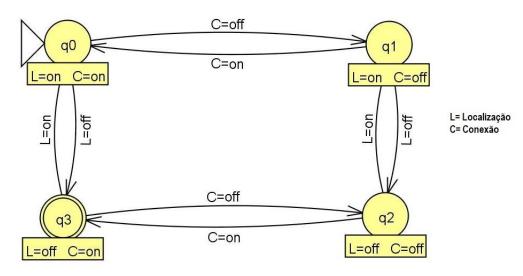


Figura 10 Máquina de estados gerada pela abordagem.

É importante ressaltar que um estado só pode ir para outro estado se e somente se houver apenas uma alteração entre as variações de elemento. Dito isso, a Tabela 2 apresenta a estrutura de casos de teste gerada pela abordagem.

Tabela 2 Casos de Teste Gerados pela Abordagem.

Campos	Exemplos				
Sequência de teste 1	Localização: on Conexão: on				

Sequência de teste 2	Localização: on Conexão: off
Sequência de teste 3	Localização: off Conexão: off
Sequência de teste 4	Localização: off Conexão: on

O testador deve inserir os comportamentos esperados para cada sequência de teste.

A Tabela 3 ilustra como seria uma inserção do resultado esperado para uma sequência de teste gerada pela abordagem para o exemplo seguido.

Campos

Sequência de teste 1

Localização: ON | Conexão: ON

Resultado Esperado

Abrir aplicativo.

Screenshot do Resultado

Esperado

Esperado

Tabela 3 Inserção dos Comportamentos Esperados.

Após a geração de casos de teste, um documento para inserção de comportamentos esperados e mais informações da aplicação a ser testada foi criado e será apresentado na seção seguinte.

3.2.2 Modelo de documento dos casos de teste

Para a criação do documento de casos teste, buscou-se seguir o padrão de casos de teste que segundo Craig e Jaskiel, (2002, apud DIAS-NETO, 2008) descreve uma condição particular a ser testada e é composto por valores de entrada, restrições para a sua execução e um resultado ou comportamento esperado. Além disso, buscou-se também fazer uma ligação com o documento de requisitos definidos na subseção 3.2. Dito isso, o documento encontra-se como mostrado no Quadro 3 e Quadro 4.

Como é mostrado no Quadro 3, os campos *Nome do aplicativo, sistema operacional* e *dispositivo*, são informações referentes a cada rodada de teste. Uma rodada de teste, como foi explicado na subseção 3.2.1, é uma rodada composta com casos de teste que serão criados para cada extração realizada em um documento de especificação de requisitos para abordagem *MTContext*.

Quadro 3 Modelo com Informações para Cada Rodada de Teste.

Campos	Exemplos			
Nome do aplicativo	Minha Bicicleta			
Plataforma do Sistema Operacional	Android			
Dispositivo	Moto x			

O Quadro 4 apresenta o modelo de caso de teste gerado para o seguinte cenário:

Ao chegar perto da escola Dom José, o aplicativo deve ativar a conexão e enviar um email com a localização do filho para a mãe.

Quadro 4 Modelo de Caso de Teste.

Campos	Exemplos				
Variaçãos do Flamento	Localização: PERTO				
Variações do Elemento	Conexão: ON				
Popultado Esparado	Enviar por email para a mãe a				
Resultado Esperado	localização atual do filho				
Screeshot do Resultado Esperado	Imagem de exemplo				
Screeshot do Resultado Esperado	Imagem de exemplo				

O campo Variações dos Elementos pode ser entendido como "Contexto", que segundo Abowd *et al.* (1999) é toda a informação passível de ser usada para caracterizar a situação de um utilizador no momento de interação. Essas variações de elementos ou Contextos são as variações geradas pelo resultado das máquinas de estado, como foi explicado na subseção 3.2.2, criadas a partir das Variações dos Elementos apresentados no Quadro 1. O campo Resultado esperado deve dizer o que se espera no comportamento da aplicação naquela determinada variação de elemento. O campo *Screenshot* foi criado por dois motivos:

- 1. Por ser visual, o testador terá facilidade na hora de comparar o resultado esperado com o resultado obtido.
- 2. Costa et al. (2015) criou novas funções em uma ferramenta denominada Sikuli. Por meio desta ferramenta, os testadores podem escrever scripts visuais que especificam quais componentes de um dispositivo móvel devem interagir e quais serão os resultados visuais esperados (CHANG, 2010). Assim, futuramente será possível verificar os padrões do campo Variações dos

Elementos, colocá-los como scripts, executá-los e ao final poder comparar o screenshot do resultado esperado com o screenshot do resultado obtido nos testes, tudo isso feito por meio da ferramenta Sikuli. Assim, o campo Screenshot foi criado para uma possível integração entre a ferramenta MTContext e a ferramenta Sikuli.

Ao final de cada rodada de testes, espera-se ter todo este modelo preenchido contendo: Nome do aplicativo, Plataforma, Dispositivo, Variações dos Elementos ou Contextos, resultado esperado e *screenshot*. Com exceção do campo *screenshot*, todos os demais são indispensáveis para se ter um caso de teste nesta abordagem proposta.

3.3 Apoio ferramental para a geração de casos de teste para Aplicações móveis Sensíveis ao contexto (*MTContext*)

A Figura 4, apresentada na subseção 3.2.1, mostra a segunda parte da abordagem em que a MTContext gera máquinas de estado e sequências de teste a partir de informações fornecidas pelo testador. Essa necessidade da abordagem em ter que gerar máquinas de estado para apoiar a geração de casos de teste para aplicações móveis sensíveis ao contexto torna a abordagem dependente de uma ferramenta que possa executar esses passos. Antes de iniciar o desenvolvimento da ferramenta, um estudo de observação foi realizado no Instituto de Desenvolvimento e Tecnologia - INDT. O estudo durou 30 dias e durante este estudo foram coletadas informações sobre como são realizados os testes para aplicações móveis sensíveis ao contexto no instituto. Além disso, foi realizada uma entrevista com dois profissionais de teste para aplicações móveis sobre como eles achavam que deveria ser ferramenta de apoio a testes sensíveis ao contexto. Logo em seguida a abordagem MTContext foi apresentada aos dois profissionais e novamente uma entrevista foi conduzida para se obter a opinião dos especialistas sobre a condução do desenvolvimento da ferramenta MTContext. Sendo assim, esta seção irá apresentar a ferramenta que implementa a abordagem e, assim, servirá de apoio a geração de casos de teste para aplicações sensíveis ao contexto.

3.3.1 Descrição da Ferramenta *MTContext*

A ferramenta *MTContext* visa prover um apoio para a geração e controle de testes para aplicações móveis sensíveis ao contexto. Seus objetivos principais são a geração de casos de testes a partir de máquinas de estados baseadas em elementos de contexto, além do controle desses testes realizados antes, durante ou após o desenvolvimento das aplicações.

MTContext está integrada a uma outra ferramenta chamada MTControol desenvolvida por Nascimento et al. (2015), que visa recomendar testes mais adequados às características, funcionalidades e plataformas de uma aplicação móvel durante a fase de planejamento dos testes, além do controle e registro da execução dos casos de teste. MTContext foi construída na linguagem PHP com o framework Yii (framework de desenvolvimento de aplicações Web que é open-source escrito em PHP 5 promovendo um design limpo que trabalha para agilizar o desenvolvimento do aplicativo), seguindo a arquitetura MVC (Model-View-Control) e utiliza o SGBD MySQL. A ferramenta está disponível em http://sistemas.icomp.ufam.edu.br/mtcontrool.

Ela é destinada a pesquisadores, desenvolvedores e testadores de aplicações móveis de grandes empresas ou individuais. Ela possui dois perfis de usuários: administrador (mantenedor do sistema e possui acesso irrestrito às suas funcionalidades) ou testador (responsável por instanciar novas rodadas de teste).

Pelo fato da *MTContext* estar integrada à ferramenta *MTControol*, algumas funcionalidades previstas para a *MTContext* já existiam na *MTControol*, assim como, algumas funcionalidades pertencentes à *MTControol* não são utilizadas pela *MTContext* e portanto, não serão mencionadas nesta dissertação.

Dito isso, nas próximas seções será apresentado como a ferramenta está estruturada para atender ao seu objetivo proposto.

3.3.2 Funcionalidades da MTContext

As funcionalidades que compõem a ferramenta *MTContext* podem ser divididas em duas categorias: (1) configuração da ferramenta e (2) apoio à realização de testes em uma aplicação móvel sensível ao contexto.

As funcionalidades que visam configurar a ferramenta são disponibilizadas a usuários com perfil de administrador. São elas:

- Gerenciar Usuários (Manage Users): CRUD⁵ de usuários que possuem acesso à ferramenta com diferentes perfis (administradores [acesso irrestrito à ferramenta] ou testadores [acesso restrito à gestão de suas aplicações]);
- Gerenciar Plataformas Móveis (*Manage Platforms*): CRUD de plataformas móveis utilizadas pelos usuários da ferramenta;

-

⁵ Acrônimo em Inglês referente às operações de Create-Retrieval-Update-Delete.

- Gerenciar Marcas (Manage Brands): CRUD de marcas (organizações/empresas) desenvolvedoras de dispositivos móveis;
- Gerenciamento de Dispositivos (Manage Devices): CRUD de dispositivos móveis utilizados por usuários da ferramenta. Tais dispositivos estão associados a plataformas móveis e marcas de dispositivos móveis.
- Gerenciamento de Elementos de Contexto (Manage Elements): CRUD que permite gerenciar os elementos de contextos utilizados pelos dispositivos. Tais elementos estão associados a dispositivos móveis e plataformas.
- Gerenciamento de Testes de Contexto (Manage Test Context): CRUD que permite gerenciar os casos de testes gerados na ferramenta. Uma vez instanciada uma nova rodada de teste, MTContext oferece uma funcionalidade de registro e controle da geração dos testes no aplicativo, de forma que todos os testes gerados estão disponíveis, bem como para quais aplicações foram gerados, plataforma, dispositivo e quais os elementos de contexto estão sendo trabalhados na rodada de teste
- Gerenciar Categorias de Aplicações (Manage Categories): CRUD que permite o gerenciamento de categorias para as aplicações da ferramenta.

As funcionalidades que visam apoiar a realização de testes em uma aplicação móvel são disponibilizadas a usuários com perfil de testador. São elas:

- Cadastrar Aplicação (New App): como entrada de dados, a ferramenta cadastra
 a aplicação móvel a ser testado, bem como todas suas propriedades (idiomas,
 categorias, plataformas, etc.).
- Criar Teste de Contexto (New Test Context): criação de uma nova instância de teste para uma aplicação já gerenciada pela ferramenta;

3.3.3 Funcionamento da MTContext

O principal diferencial da ferramenta é o fato de gerar contextos de forma automatizada após a inserção de requisitos explícitos da aplicação móvel. Esses contextos são baseados no agrupamento de todas as possíveis variações entre todos os elementos cadastrados em um teste. É importante ressaltar que estes contextos gerados podem informar situação não esperadas pelos desenvolvedores da aplicação, bem como situações previamente esperadas. Ao longo desta seção serão descritas as principais funcionalidades da *MTContext*.

3.3.3.1 Cadastro de Uma Aplicação Móvel

Para iniciar o uso da ferramenta, um usuário com perfil de testador precisa inicialmente cadastrar uma aplicação móvel, como mostrado na Figura 11. Nesse momento, o usuário informa as principais propriedades da aplicação (ex: descrição, idiomas, categorias, plataformas).

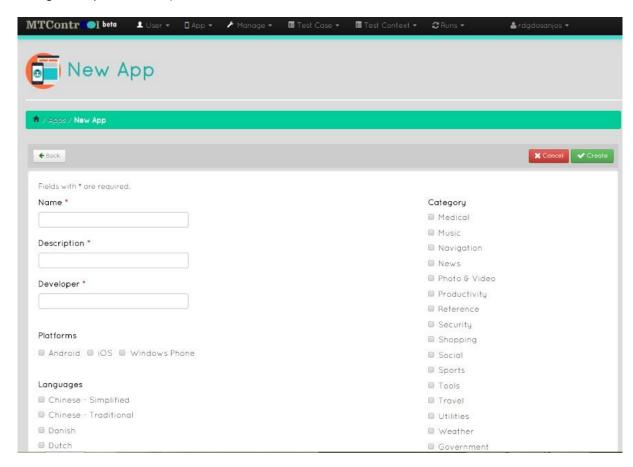


Figura 11 Tela de Cadastro de Aplicação Móvel.

Uma aplicação pode ser associada a diversas plataformas, não necessitando a criação de várias instâncias de uma mesma aplicação para várias plataformas.

3.3.3.2 Cadastro de um Dispositivo Móvel

Existe um cadastro de dispositivo móvel (*device*), pois com os avanços tecnológicos em hardware, é comum vermos lançamentos de dispositivos móveis todo ano para diferentes plataformas. Assim, essa funcionalidade é necessária para que se possa ter um controle sobre os dispositivos móveis existentes para cada plataforma.

A Figura 12 apresenta o cadastro de um elemento. Neste cadastro é preciso selecionar qual a Organização produtora do dispositivo (as informações desse campo são coletadas por meio da funcionalidade "cadastrar marca de dispositivo móvel" (*Create Brand*),

selecionar qual a plataforma do dispositivo (proveniente das informações inseridas em" cadastrar plataforma") e o seu modelo (no exemplo abaixo, *Moto Force*). Essas informações serão muito importantes no momento de se cadastrar um elemento de contexto e isso será apresentado na seção seguinte.

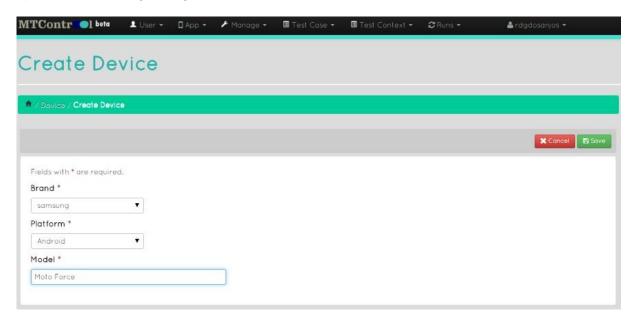


Figura 12 Tela de Cadastro de Dispositivo Móvel.

3.3.3.3 Cadastro de um Elemento

O cadastro de um elemento de contexto é divido em duas fases: (a) cadastrar elemento e a qual plataforma esse elemento pertence e (b) configurar a quais dispositivos este elemento pertence.

A Figura 13 apresenta a tela de cadastro de um elemento na *MTContext*. Nesta etapa é preciso cadastrar qual o nome do ELEMENTO, no exemplo o nome é "temperatura". Em seguida, deve-se informar em quais plataformas este elemento está presente. É possível indicar mais de uma plataforma.

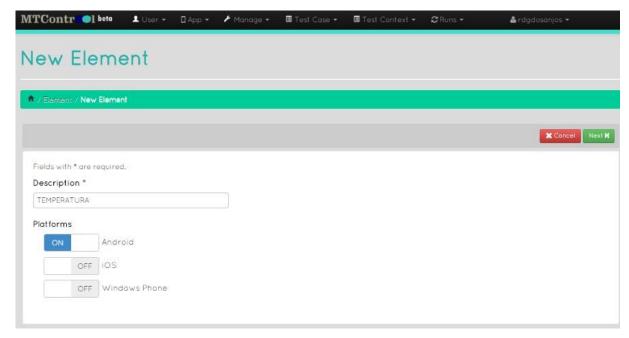


Figura 13 Tela de cadastro de um elemento e suas plataformas (a).

A Figura 14 apresenta a segunda etapa do cadastro de um elemento. Nesta etapa serão listados todos os dispositivos cadastrados para as plataformas indicadas na etapa (a) e é necessário indicar quais desses dispositivos possuem o elemento em sua configuração. Isto acontece, pois é preciso levar em consideração o fato de que alguns dos dispositivos são superiores e inferiores a outros. Ou seja, alguns dispositivos de determinadas plataformas podem utilizar um elemento (como no exemplo, "Temperatura") que outros por serem inferiores ou não possuem tal configuração ou não a utilizam.

3.3.3.4 Geração de um teste

A funcionalidade geração deve ser feita após extração das informações contidas no documento *reqContext*. Para cada caso de uso ou cenário de uma aplicação deve ser preenchido um formulário do *reqContext*, e para cada formulário uma geração de teste deve ser criada.

Para criar um teste para aplicações móveis sensíveis ao contexto na *MTContext*, primeiro deve-se informar o nome do teste que será feito. Como boa prática, é indicado colocar um nome que identifique o cenário ou o caso de uso a ser testado. Logo após, deve ser selecionada a aplicação que será testada. Essa aplicação deve ter sido previamente cadastrada na ferramenta. Logo em seguida, deve-se informar em qual plataforma o teste será realizado. Finalmente, deve-se informar em qual dispositivo ocorrerá o teste (Figura 15).

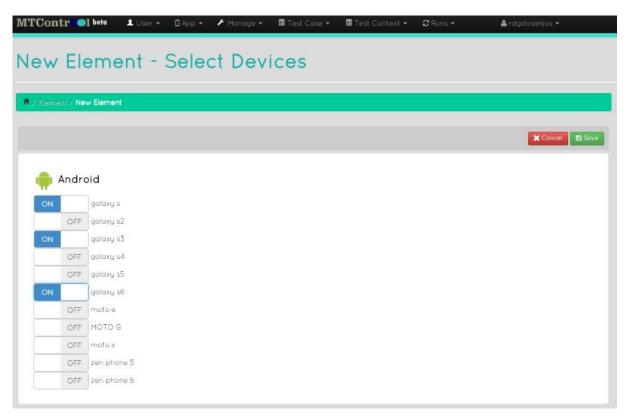


Figura 14 Tela de cadastro dos dispositivos associados a um elemento (b).

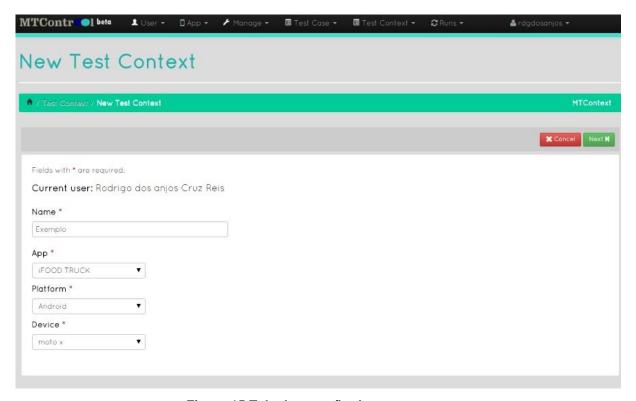


Figura 15 Tela de geração de um teste.

A Figura 16 apresenta a tela de seleção dos elementos de contexto presentes neste cenário de teste. Nesta etapa, serão listados todos os elementos cadastrados previamente como pertencentes a esta plataforma e este dispositivo que foram selecionados para o teste.

Além disso, o usuário deve escolher qual o tipo do elemento, se é nominal ou intervalar, como foi explicado na subseção 3.2.1.

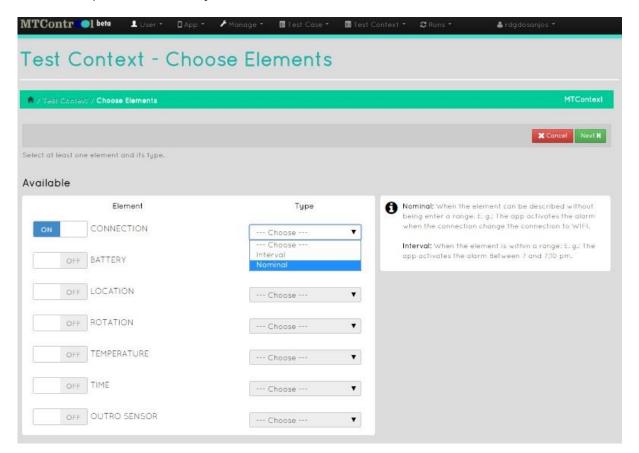


Figura 16 Escolha dos Elementos de Contexto.

A Figura 17 apresenta a tela de instanciação dos elementos. Nesta etapa o testador deve inserir todas as variações existentes dos elementos conforme extraídos no documento de especificação de requisitos. Após ter inserido todas as instâncias dos elementos, ou seja, ter inserido todas as variações dos elementos, a ferramenta irá redirecionar a página para uma tela de resumo contendo todas as informações já cadastradas até o momento para que o usuário possa avaliar se está tudo como planejado ou faltou alguma informação e assim adicioná-la ou corrigi-la. Estando tudo certo, o usuário então precisará gerar as sequências de teste. Ao selecionar a opção para gerar as sequências de teste, todo o processo explicado na subseção 3.2.2, Estrutura das Máquinas de Estados e Geração de Sequências de Testes, ocorrerá.

Após o processo de geração de máquinas de estado e geração das sequências de teste, uma tela com todas as sequências geradas será apresentada, como mostra a Figura 18. Nesta tela, o usuário poderá visualizar a variação ou novo contexto gerado, avaliá-lo se realmente faz parte do seu cenário e então poderá inserir qual o comportamento esperado

para aquele determinado contexto e inserir qual o *screenshot* deste comportamento, caso exista.

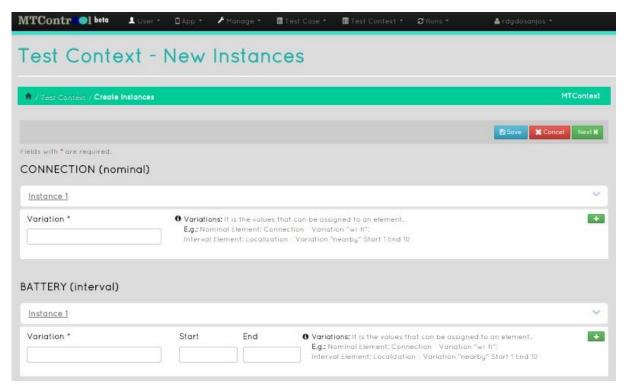


Figura 17 Tela de geração de um teste - Nova Instância.

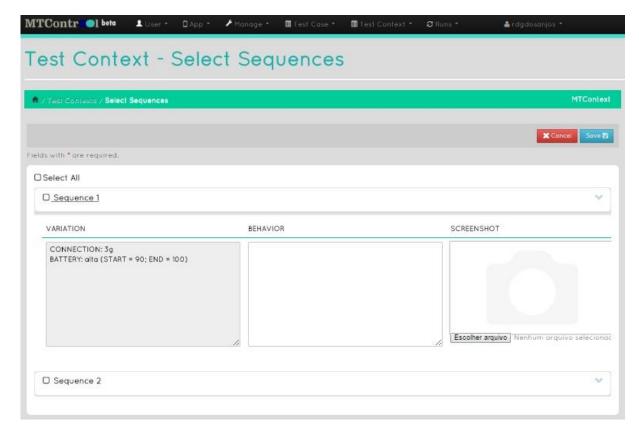


Figura 18 Inserção de Comportamento Esperado.

3.3.3.5 Controle dos Testes

Após o preenchimento de todos os comportamentos esperados e suas *screenshots*, o testador pode visualizar todas as informações do teste realizado. A tela *Dashboard* apresenta todos os casos de teste, juntamente com seus comportamentos esperados cadastrados. Além disso, o usuário pode também ter a informação de quem gerou esse teste e quais foram as informações cadastradas para este teste.

A Figura 19 apresenta a tela Dashboard que permite ao usuário consultar quais as sequências de testes foram geradas para determinada aplicação. Além disso, o testador pode gerenciar todos os testes gerados para determinada aplicação ou todos os testes gerados por determinado usuário, plataforma ou dispositivo por meio do item *Manage Test Context*.

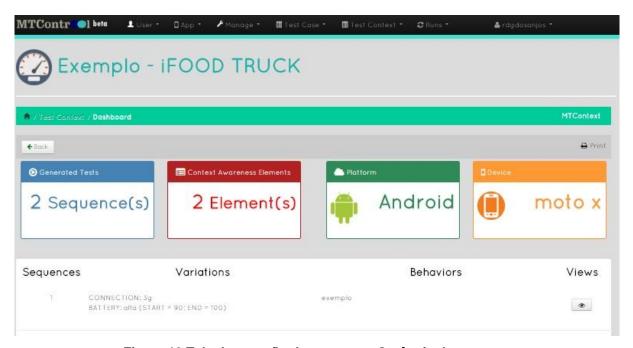


Figura 19 Tela de geração de um teste - Gerência do teste.

No próximo capítulo, será apresentado um estudo que teve como objetivo avaliar a viabilidade da abordagem por meio da utilização da ferramenta *MTContext* em relação à geração de casos de testes reais para aplicações móveis sensíveis ao contexto.

CAPÍTULO 4 - ESTUDO DE VIABILIDADE PARA AVALIAÇÃO DA FERRAMENTA MTCONTEXT

Este capítulo descreve o estudo de viabilidade realizado seguindo uma metodologia científica para avaliação de tecnologias de software, que consiste em duas seções de geração de teste para aplicações móveis sensíveis ao contexto, sendo uma utilizando a abordagem ad hoc e a outra seção utilizando a ferramenta MTContext a fim de comparar os resultados das seções e avaliar eficiência e eficácia da abordagem.

4.1. Introdução

Devido à importância dos testes de software no processo de desenvolvimento e principalmente na qualidade do produto final, a avaliação da qualidade da app por meio de suas interfaces produzidas ao longo do processo de desenvolvimento, verificando se elas atendem aos requisitos especificados pelo cliente e projetado pela equipe técnica do projeto, passa a ter extrema importância para os *stakeholders* (profissionais envolvidos diretamente e indiretamente no projeto). Neste contexto, foi proposta uma abordagem para testes em aplicações móveis sensíveis ao contexto. Essa abordagem foi implementada na ferramenta *MTContext*, visando gerar casos de teste para diferentes contextos existentes em uma aplicação móvel que possam identificar erros nas *apps* por meio de comparação entre resultado esperado e resultado obtido das telas da aplicação.

Após a construção da *MTContext*, fez-se necessária a realização de estudos de avaliação visando caracterizar essa nova abordagem em relação à sua capacidade em gerar casos de testes e visando permitir futuramente a transferência dessa tecnologia para o contexto industrial, reduzindo os riscos para a implantação. Para isso, a metodologia definida por SHULL et al. (2001) foi utilizada. Ela é composta por quatro etapas, das quais foi aplicada nesta pesquisa apenas uma, devido ao tempo necessário para a realização de todas as etapas serem longos não podendo ser realizado durante esta pesquisa de mestrado.

Assim, o estudo realizado foi o estudo de viabilidade e teve como objetivo comparar a execução da *MTContext* com a forma Ad Hoc em relação a eficiência e eficácia para identificação para geração de casos de teste em um app. Seu planejamento, projeto e resultados estão descritos a seguir.

4.2. Definição do Estudo de Viabilidade

O objetivo do estudo de viabilidade é fornecer ao pesquisador informações suficientes para justificar (ou não) a continuação do seu trabalho (SHULL, 2001).

4.2.1. Propósito

O propósito do estudo foi responder a questão:

"A aplicação da abordagem *MTContext* por meio de uma ferramenta para geração de casos de teste para aplicações móveis sensíveis ao contexto é viável, analisando sua eficácia, eficiência e cobertura de variações de elementos e contextos?"

4.2.2. Perspectiva

A perspectiva é do ponto de vista dos pesquisadores, que desejam analisar a viabilidade da aplicação da ferramenta proposta.

4.2.3. Objetivos específicos

- Analisar a utilização da forma automatizada MTContext e a forma tradicional Ad hoc na geração de casos de teste;
- Com o propósito de caracterizar;
- o **Com respeito a** eficácia, eficiência e cobertura das formas em gerar casos de teste;
- Do ponto de vista do pesquisador;
- No contexto de geração de casos de teste para aplicações móveis sensíveis ao contexto.

4.2.4. Questões e Métricas

As questões abaixo se referem à aplicação das abordagens de geração de casos de teste em aplicações móveis sensíveis ao contexto, *MTContext e Ad Hoc.* A saber:

 Q1: Qual é a eficácia da utilização das abordagens MTContext e Ad Hoc no que diz respeito à geração de casos de teste?

Métricas:

- CT_(mtcontext) = Quantidade de casos de teste gerados pela abordagem MTContext.
- o CT_(adhoc) = Quantidade de casos de teste gerados pela abordagem Ad Hoc.
- CT_{v(mtcontext)} = Quantidade de casos de teste válidos gerados pela abordagem *MTContext*.

- \circ CT_{v (adhoc)} = Quantidade de casos de teste válidos gerados pela abordagem Ad Hoc.
- CT_{iv(mtcontext)} = Quantidade de casos de teste inválidos gerados pela abordagem *MTContext*.
- CT_{iv (adhoc)} = Quantidade de casos de teste inválidos gerados pela abordagem Ad
 Hoc
- Eficácia-a_(mtcontext) = CT_{v(mtcontext)} * 100/ CT_(mtcontext).
- Eficácia- $a_{(adhoc)} = CT_{v(adhoc)} * 100/ CT_{(adhoc)}$.
- Eficácia- $b_{(mtcontext)} = CT_{iv(mtcontext)} * 100/ CT_{(mtcontext)}$
- \circ Eficácia-b_(adhoc) = CT_{iv(adhoc)} * 100/ CT_(adhoc).
- Q2: Qual a cobertura da aplicação das abordagens MTContext e Ad Hoc no que diz respeito à geração de casos de teste?

Métricas:

- C_(mtcontext) = Quantidade de contextos alcançados pela abordagem MTContext.
- o C_(adhoc) = Quantidade de contextos alcançados pela abordagem *Ad hoc*.
- TC = Total de contextos possíveis (o mesmo em ambas as abordagens).
- o TV = Total de variações possíveis (o mesmo em ambas as abordagens).
- V_(mtcontext) = Quantidade de Variações de contextos alcançadas pela abordagem *MTContext*.
- V_(adhoc) = Quantidade de Variações de contextos alcançadas pela abordagem Ad Hoc.
- Cobertura-a_(mtcontext) = C *100 / TC_(mtcontext).
- \circ Cobertura-a_(adhoc) = C *100 / TC_(adhoc).
- o Cobertura- $b_{(mtcontext)} = V *100 / TV_{(mtcontext)}$.
- \circ Cobertura-b_(adhoc) = V *100 / TV_(adhoc)
- Q3: Qual é a eficiência da aplicação das abordagens MTContext e Ad Hoc no que diz respeito à geração de casos de teste?

Métricas:

- o T_(mtcontext) = Tempo de execução da *MTContext*.
- T_(adhoc) = Tempo de execução da AdHoc.
- Eficiência-a_(mtcontext) = CT_(mtcontext) / T_(mtcontext).
- Eficiência-a_(adhoc) = CT_(adhoc). / T_(adhoc).
- \circ Eficiência-b_(mtcontext) = C_(mtcontext) / T_(mtcontext)
- o Eficiência- $b_{(adhoc)} = C_{(adhoc)} / T_{(adhoc)}$.

- o Eficiência- $c_{(mtcontext)} = V_{(mtcontext)} / T_{(mtcontext)}$
- o Eficiência- $c_{(adhoc)} = V_{(adhoc)} / T_{(adhoc)}$

4.3. Planejamento do Estudo

Esse estudo teve o objetivo de geração de casos de teste de aplicações móveis por meio da utilização de duas abordagens de geração de testes: *MTContext* e Ad hoc. Com os resultados obtidos a partir da realização desse estudo é possível comparar a abordagem *MTContext* em relação aos resultados obtidos com a abordagem Ad Hoc. Com isso, neste estudo, buscou-se adquirir conhecimento que permita avaliar a viabilidade de aplicação da abordagem proposta.

Com os resultados obtidos neste estudo e o corpo de conhecimento construído decorrente de sua condução, espera-se que forneça subsídios que permitam evoluir a abordagem para otimizar a sua aplicação tanto em relação ao esforço necessário para executá-la quanto à quantidade e tipo de casos de teste que ela possa gerar.

4.3.1. Formulação de Hipóteses

Para avaliação, foi realizada uma comparação entre a abordagem automatizada *MTContext* e a abordagem tradicional, Ad Hoc. Essa avaliação foi feita de acordo com as métricas citadas anteriormente analisando o número de casos de teste, percentual de casos de testes válidos e inválidos gerados pelas abordagens, percentual de contextos e variações de contextos cobertas pelos casos de teste gerados por ambas as abordagens.

Deseja-se comparar a eficácia, eficiência e cobertura das abordagens *MTContext* e *Ad Hoc* na geração de casos de teste para aplicações móveis sensíveis ao contexto. A eficácia, eficiência e cobertura são medidas pelas métricas citadas na subseção 4.2.4.

Hipótese nula (H0): Não existe diferença entre os resultados encontrados pela abordagem *MTContext* e pela abordagem Ad Hoc em relação a geração de casos de teste e cobertura de contextos e variações em aplicações móveis.

Para se obter resultados com o objetivo de refutar ou confirmar a hipótese H0, a eficácia, eficiência e cobertura das abordagens foram as características avaliadas. Portanto, para cada uma dessas duas características, as seguintes hipóteses foram definidas:

Eficácia:

- CT_(mtcontext) = Quantidade de casos de teste gerados pela abordagem MTContext.
- CT_(adhoc) = Quantidade de casos de teste gerados pela abordagem Ad Hoc.

• **Hipótese nula A** (H0 A): A abordagem *MTContext* permite gerar a mesma quantidade de casos de teste gerados pela abordagem *Ad Hoc*.

H0 A:
$$CT_{(mtcontext)} = CT_{(adhoc)}$$

 Hipótese Alternativa 1 A (H1 A): A abordagem MTContext permite gerar mais casos de teste que aqueles encontrados pela abordagem Ad Hoc.

H1 A:
$$CT_{(mtcontext)} > CT_{(adhoc)}$$

 Hipótese Alternativa 2 A (H2 A): A abordagem MTContext permite gerar menos casos de teste que aqueles encontrados pela abordagem Ad Hoc.

H2 A:
$$CT_{(mtcontext)} < CT_{(adhoc)}$$

- CT_{v(mtcontext)} = Quantidade de casos de teste válidos gerados por MTContext.
- CT_{v(adhoc)} = Quantidade de casos de teste válidos gerados por Ad Hoc.
- CT_{v(mtcontext)} * 100/ CT_(mtcontext) = Percentual de Casos de tese válidos dividido pela quantidade de casos de teste gerados pela MTContext.
- CT_{v(adhoc)} * 100/ CT_(adhoc) = Percentual de Casos de tese válidos dividido pela quantidade de casos de teste gerados pela abordagem *Ad Hoc*.
- Hipótese nula B (H0 B): A abordagem MTContext permite gerar o mesmo percentual de casos de testes válidos gerados pela abordagem Ad Hoc.

H0 A:
$$CT_{v(mtcontext)} * 100/ CT_{(mtcontext)} = CT_{v(adhoc)} * 100/ CT_{(adhoc)}$$

 Hipótese Alternativa 1 B (H1 B): A abordagem MTContext permite gerar um percentual maior de testes válidos que aqueles gerados pela abordagem Ad Hoc.

 Hipótese Alternativa 2 B (H2 B): A abordagem MTContext permite gerar um percentual menor de casos de teste válidos que aqueles gerados pela abordagem Ad Hoc.

- CT_{iv(mtcontext)} = Quantidade de casos de teste inválidos gerados pela abordagem MTContext.
- CT_{iv (adhoc)} = Quantidade de casos de teste inválidos gerados pela abordagem Ad Hoc.
- CT_{iv(mtcontext)} * 100/ CT_(mtcontext) = Percentual de Casos de tese inválidos dividido pela quantidade de casos de teste gerados pela abordagem *MTContext*.

- CT_{iv(adhoc)} * 100/ CT_(adhoc) = Percentual de Casos de tese inválidos dividido pela quantidade de casos de teste gerados pela abordagem Ad Hoc.
- Hipótese nula B (H0 B): A abordagem MTContext permite gerar o mesmo percentual de casos de testes inválidos gerados pela abordagem Ad Hoc.

• **Hipótese Alternativa 1 B (H1 B)**: A abordagem *MTContext* permite gerar um percentual maior de testes inválidos que aqueles gerados pela abordagem *Ad Hoc*.

 Hipótese Alternativa 2 B (H2 B): A abordagem MTContext permite gerar um percentual menor de casos de teste inválidos que aqueles gerados pela abordagem Ad Hoc.

Cobertura

- C_(mtcontext) = Quantidade de contextos alcançados pela abordagem MTContext.
- C_(adhoc) = Quantidade de contextos alcançados pela abordagem *Ad hoc*.
- TC = Total de contextos possíveis (o mesmo em ambas as abordagens).
- C *100 / TC_(mtcontext) = Percentual de contextos cobertos pela abordagem MTContext.
- C *100 / TC_(adhoc) = Percentual de contextos cobertos pela abordagem *Ad Hoc*.
- Hipótese nula C (H0 C): A abordagem MTContext permite cobrir o mesmo percentual de contextos que a abordagem Ad Hoc.

H0 C: C *100 /
$$TC_{(mtcontext)} = C *100 / $TC_{(adhoc)}$$$

 Hipótese Alternativa 1 C (H1 C): A abordagem MTContext permite cobrir um percentual maior de contextos que a abordagem Ad Hoc.

H1 C: C *100 /
$$TC_{(mtcontext)} > C *100 / $TC_{(adhoc)}$$$

 Hipótese Alternativa 2 C (H2 C): A abordagem MTContext permite cobrir um percentual menor de contextos que a abordagem Ad Hoc.

H2 C: C *100 /
$$TC_{(mtcontext)}$$
 < C *100 / $TC_{(adhoc)}$

- TV = Total de variações possíveis (o mesmo em ambas as abordagens).
- V_(mtcontext) = Quantidade de Variações de contextos alcançadas pela abordagem MTContext.
- V_(adhoc) = Quantidade de Variações de contextos alcançadas pela abordagem Ad Hoc.
- V_(mtcontext) *100 / TV = Percentual de variações de contextos cobertos pela abordagem MTContext
- V_(adhoc) *100 / TV = Percentual de variações de contextos cobertos pela abordagem Ad Hoc.
- **Hipótese nula D** (H0 D): A abordagem *MTContext* permite cobrir o mesmo percentual de contextos que a abordagem *Ad Hoc*.

H0 D:
$$V_{(mtcontext)}$$
 *100 / TV = $V_{(adhoc)}$ *100 / TV

 Hipótese Alternativa 1 D (H1 D): A abordagem MTContext permite cobrir um percentual maior de contextos que a abordagem Ad Hoc.

H1 D
$$V_{(mtcontext)}$$
 *100 / TV > $V_{(adhoc)}$ *100 / TV

 Hipótese Alternativa 2 D (H2 D): A abordagem MTContext permite cobrir um percentual menor de contextos que a abordagem Ad Hoc.

Eficiência

- T_(mtcontext) = Tempo de execução da abordagem *MTContext*.
- T_(adhoc) = Tempo de execução da abordagem Ad Hoc.
- Hipótese nula E (H0 E): O tempo para a Geração dos casos de teste a abordagem
 MTContext é o mesmo que o Tempo para a geração de casos de teste da
 abordagem *Ad Hoc*.

H0 E:
$$T_{(mtcontext)} = T_{(adhoc)}$$

 Hipótese Alternativa 1 E (H1 E): O tempo para a Geração dos casos de teste a abordagem MTContext é maior que o Tempo para a geração de casos de teste da abordagem Ad Hoc.

H1 E:
$$T_{(mtcontext)} > T_{(adhoc)}$$

 Hipótese Alternativa 2 E (H2 E): O tempo para a Geração dos casos de teste a abordagem MTContext é menor que o Tempo para a geração de casos de teste da abordagem Ad Hoc.

H2 E: $T_{(mtcontext)} < T_{(adhoc)}$

- CT_(mtcontext) / T_(mtcontext) = Quantidade de casos de teste gerados pela abordagem
 MTContext dividido pelo tempo da abordagem MTContext.
- CT_{(adhoc).} / T_(adhoc) = Quantidade de casos de teste gerados pela abordagem Ad
 Hoc dividido pelo tempo da abordagem Ad Hoc.
- Hipótese nula F (H0 F): A relação entre a quantidade de casos de teste válidos gerados pelo tempo de execução é a mesma para as abordagens MTContext e Ad Hoc.

H0 C:
$$CT_{v(mtcontext)} \div T_{(mtcontext)} = CT_{v(adhoc)} \div T_{(adhoc)}$$

 Hipótese Alternativa 1 F (H1 F): A relação entre a quantidade de casos de teste válidos gerados pelo tempo de execução utilizando a abordagem MTContext é maior que o da abordagem Ad Hoc.

H1 C:
$$CT_{v(mtcontext)} \div T_{(mtcontext)} > CT_{v(adhoc)} \div T_{(adhoc)}$$

 Hipótese Alternativa 2 F (H2 F): A relação entre a quantidade de casos de teste válidos gerados pelo tempo de execução utilizando a abordagem MTContext é menor que o da abordagem Ad Hoc.

H2 C
$$CT_{v(mtcontext)} \div T_{(mtcontext)} < CT_{v(adhoc)} \div T_{(adhoc)}$$

- C_(mtcontext) / T_(mtcontext) = Cobertura de Contextos cobertos dividido pelo tempo da abordagem MTContext.
- C_(adhoc) / T_(adhoc) = Cobertura de Contextos cobertos dividido pelo tempo da abordagem Ad Hoc.
- **Hipótese nula C (H0 C):** A relação entre a quantidade de contextos cobertos pelo tempo de execução é a mesma para as abordagens *MTContext* e *Ad Hoc*.

H0 C:
$$C_{(mtcontext)} \div T_{(mtcontext)} = C_{(adhoc)} \div T_{(adhoc)}$$

 Hipótese Alternativa 1 C (H1 C): A relação entre a quantidade de contextos cobertos pelo tempo de execução utilizando a abordagem MTContext é maior que o da abordagem Ad Hoc.

H1 C:
$$C_{(mtcontext)} \div T_{(mtcontext)} > C_{(adhoc)} \div T_{(adhoc)}$$

Hipótese Alternativa 2 C (H2 C): A relação entre a quantidade contextos cobertos e
o tempo de execução utilizando a abordagem MTContext é menor que o da
abordagem Ad Hoc.

H2 C:
$$C_{(mtcontext)} \div T_{(mtcontext)} < C_{(adhoc)} \div T_{(adhoc)}$$

- V_(mtcontext) / T_(mtcontext) = Variações de contexto cobertas pela *MTContext* dividido pelo tempo da *MTContext*.
- V_(adhoc) / T_(adhoc) = Variações de contexto cobertas pela Ad Hoc dividido pelo tempo da Ad Hoc.
- Hipótese nula C (H0 C): A relação entre a quantidade de variações de contextos cobertas e o tempo de execução é a mesma para as abordagens MTContext e Ad Hoc.

H0 C:
$$V_{(mtcontext)} \div T_{(mtcontext)} = V_{(adhoc)} \div T_{(adhoc)}$$

 Hipótese Alternativa 1 C (H1 C): A relação entre a quantidade de variações de contextos cobertas e o tempo de execução utilizando a abordagem MTContext é maior que o da abordagem Ad Hoc.

H1 C:
$$V_{(mtcontext)} \div T_{(mtcontext)} > V_{(adhoc)} \div T_{(adhoc)}$$

 Hipótese Alternativa 2 C (H2 C): A relação entre a quantidade de variações de contextos cobertas e o tempo de execução utilizando a abordagem MTContext é menor que o da abordagem Ad Hoc.

H2 C:
$$V_{(mtcontext)} \div T_{(mtcontext)} < V_{(adhoc)} \div T_{(adhoc)}$$

4.3.2. Seleção dos participantes

Os participantes deste estudo foram provenientes da lista de email de alunos e egressos dos cursos de graduação e pós-graduação do Instituto de Computação (IComp) da Universidade Federal do Amazonas - UFAM. Entretanto, para participar do estudo, os estudantes tiveram que:

Manifestar interesse em participar do estudo, assinando o Formulário de Consentimento (

 Apêndice A Termo de Consentimento livre e Esclarecido – TCLE) e formulário de caracterização (Apêndice B - Formulário de Caracterização), para termos conhecimento do grau de experiência de cada aluno:

- Participar do treinamento que foi conduzido como uma explicação sobre Sensibilidade ao contexto, geração de teste para aplicações móveis sensíveis ao contexto, utilização da abordagem MTContext e quaisquer dúvidas por parte dos alunos.
- Terem cursado alguma disciplina que abordou o tema teste de software ou terem participado de projetos reais atuando em tarefas relacionadas a teste de software, pois durante esse experimento será necessário algum conhecimento prévio sobre teste de software.

4.3.3. Seleção de Grupos

Cada aluno (testador) foi analisado e avaliado individualmente, por meio do questionário de caracterização (Apêndice B) que os mesmos preencheram com informações que permitiram identificar o grau de conhecimento, experiência e competência, tendo em vista que são alunos de graduação em estágio final e alunos de pós-graduação. Para este estudo, os mesmos foram organizados em dois grupos. Para compor os grupos, certas precauções foram tomadas visando reduzir eventuais fatores de confusão, por conta do nível de qualificação dos participantes. Para isso, os grupos foram mesclados de acordo com as informações contidas no formulário de caracterização.

4.3.4. Aplicações a serem testadas

Foram especificadas 2 (duas) aplicações não reais (não desenvolvidas) para que os participantes pudessem gerar casos de teste para elas. Essas aplicações foram escolhidas por conterem descrições explícitas de características de sensibilidade ao contexto. Essas aplicações são apresentadas por meio de um documento contendo descrição dos requisitos da aplicação. São elas, iFOOD TRUCK (Apêndice C - Requisitos das Aplicações - APP 2 - iFOOD TRUCK) e Minha Bicicleta (Apêndice D - Requisitos das Aplicações - APP 1 - MINHA BICICLETA). Além da descrição, estes documentos possuem três cenários que foram especificados para uso durante a execução do estudo que será apresentado na seção seguinte.

4.4. Ameaças à Validade do Estudo

Nesta seção serão detalhadas as possíveis ameaças à validade dos resultados.

4.4.1. Validade Interna

Ameaças à validade interna podem indicar uma relação causal inexistente entre tratamento e resultado. As principais ameaças identificadas foram:

- Instrumentação: os formulários a serem utilizados no estudo experimental podem influenciar de forma significativa a condução do estudo, afetando negativamente seus resultados. Entretanto, por tratar-se de um estudo de viabilidade, acredita-se que essa questão não representa uma ameaça significativa à validade do estudo.
- Seleção: a amostra dos participantes não foi aleatória, visto que será proveniente da disponibilidade de alunos em uma disciplina que abordou o tema teste de software na UFAM, e os demais são profissionais da indústria.
- Plágio: um típico risco na condução de estudos experimentais em ambiente acadêmico é a potencial troca de informações entre os participantes sobre as tarefas do estudo. Entretanto, acredita-se que essa ameaça não é pertinente, visto que os participantes se comprometeram a não se comunicarem durante a realização do estudo e o resultado da aplicação da abordagem não resultará em uma avaliação para a disciplina.

4.4.2. Validade Externa

Ameaças à validade externa estão relacionadas com a capacidade de generalizar resultados do experimento fora do seu ambiente. As principais ameaças identificadas foram:

- Interação entre Participante e Tratamento: A amostra dos participantes selecionados é bastante heterogênea. Boa parte dos alunos possui experiência e ainda atua em ambiente industrial e a outra parte já participou pelo menos uma vez de um projeto acadêmico em que realizou teste de software.
- Interação entre Ambiente e Tratamento: um ambiente acadêmico não consegue simular as condições existentes em um ambiente industrial completamente. Porém, por utilizar documentos de teste criados também em um ambiente acadêmico, essa ameaça pode não ser considerada.

4.4.3. Validade de Construção

Ameaças à validade de construção referem-se à extensão que a configuração do experimento reflete na construção em estudo. A ameaça identificada foi:

- Viés mono-operação: o estudo de viabilidade utilizou duas aplicações como artefatos de teste. O estudo pode não ser representativo para outra pode não ficar claro se a causa dos resultados do estudo é decorrente da aplicação de MTContext ou da facilidade de geração de testes nos documentos de requisitos das aplicações utilizadas.
- Teste: a fim de evitar influências no comportamento do participante, o objetivo do
 estudo não foi apresentado aos participantes. Além disso, os participantes não foram
 envolvidos com discussão sobre as supostas vantagens e desvantagens da
 utilização das abordagens. Porém, esta ameaça não está totalmente descartada.

4.4.4. Validade de Conclusão

Ameaças à validade de conclusão estão preocupadas com questões que afetam a capacidade de tirar a conclusão correta sobre as relações entre o tratamento e o resultado.

- Confiança das Medidas: a fim de reduzir esta ameaça a validade, a apresentação dos resultados em um seminário ao grupo EXPERTS foi adotada. Além disso, todos os dados coletados estão expostos nesta dissertação e podem facilmente serem reproduzidos.
- Heterogeneidade Aleatória dos Participantes: os participantes foram egressos e alunos de graduação em Sistemas e pós-graduação em Engenharia de software.
 Portanto, não se esperam participantes com níveis de competências tão discrepantes que possam a vir comprometer a validade do estudo.

4.5. Execução do Estudo

O estudo foi conduzido de forma assíncrona e remota, ou seja, nem todos os participantes realizaram os testes ao mesmo tempo e nem no mesmo lugar. O estudo foi realizado em duas sessões. Antes da primeira sessão, todos os participantes foram orientados a preencher um Formulário de Consentimento e receberam todas as informações acerca do estudo, das atividades que deveriam ser realizadas e como deveriam proceder. Após o preenchimento do Formulário de Consentimento, os participantes responderam ao Formulário de Caracterização visando identificar a competência e a experiência relacionadas ao propósito do estudo, além de se buscar identificar o conhecimento do participante no domínio de teste em aplicações móveis.

O estudo foi realizado em ambiente acadêmico, então para estimular o interesse dos alunos a participarem do estudo, foi divulgado que ao final do estudo os alunos receberiam um certificado com 6 (seis) horas por participação no estudo. Porém, para os egressos, o

interesse em colaborar com a pesquisa deveria partir do próprio aluno, não havendo nenhum tipo de bonificação por participação. Assim, 10 alunos manifestaram interesse em participar, sendo 6 egressos e 4 alunos correntes. Estes foram divididos em dois grupos com (5) alunos cada, compondo as equipes A e B (Figura 20).

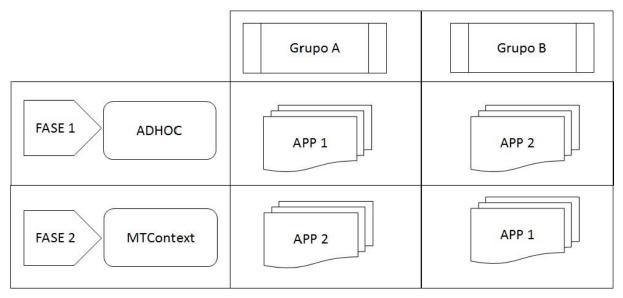


Figura 20 Estratégia de Execução do Estudo.

O estudo consistiu na simulação de uma sessão de teste de software. Os participantes em um primeiro momento aplicaram a técnica *Ad Hoc* nas aplicações móveis. Cada participante recebeu individualmente o documento com requisitos da aplicação associadas a seu grupo e os cenários a serem testados. Foi entregue também um documento para auxílio na identificação dos elementos de contexto presente nos requisitos uma para a APP 1 (Apêndice E - Roteiro do Participante – 1° Sessão APP1 – Minha Bicicleta) e ouro para APP 2 (Apêndice F - Roteiro do Participante – 1° Sessão APP2 – iFOODTRUCK) e um documento para preenchimento com os casos de teste gerados (Apêndice G - Documento para a especificação dos casos de teste AD HOC). Neste momento, os membros da equipe A receberam a aplicação A1 e os membros da equipe B receberam a aplicação A2. Após a sessão de teste, os participantes das equipes (A e B) entregaram o documento com os casos de teste preenchidos, ficando assim aptos a receber o treinamento para segunda etapa do projeto.

Após a entrega da lista de casos de testes gerados (pré-requisito para próxima etapa), os alunos receberam um treinamento para utilização da segunda abordagem, *MTContext*. Eles foram treinados a respeito da utilização dos *procedimentos da ferramenta* e tentou-se eliminar dúvidas por parte dos alunos para que fosse possível sua aplicação sem intervenção do pesquisador. Após o treinamento, as aplicações foram distribuídas entre os grupos.

Nesse momento foram invertidas as aplicações, de forma que os participantes do Grupo A testassem a aplicação A2, enquanto que os participantes do Grupo B testassem a aplicação A1, ambos utilizando a ferramenta *MTContext* como abordagem de apoio.

A dinâmica das duas fases está apresentada na Figura 20. Não foi imposto um limite de tempo para que ambas as avaliações fossem realizadas. No primeiro contato com os alunos, houve a explicação de algumas informações, dentre elas foi pedido um máximo de atenção em relação ao tempo gasto durante todo o período da sessão de teste, ficando acordado que ao iniciar e ao terminar a sessão o horário deveria ser rigorosamente anotado, assim como o horário de início e fim, o número de casos de teste gerados pela ferramenta e o número de casos de teste escolhidos pelo usuário para cada cenário da aplicação. Para isso foi entregue um documento que conteriam esses campos para serem registrados (**Apêndice H - Roteiro do Participante – 2º Sessão**).

Após a conclusão do experimento, os participantes exportaram todos os casos de teste gerados na ferramenta e enviaram por email ao condutor do estudo. Após o envio, os participantes responderam a um formulário de avaliação pós experimento (**Apêndice I - Avaliação Pós Experimento - Teste para AMSC**) com o intuito de coletar informações para levantamento qualitativo de uso das abordagens. Desta forma não houve benefício nem para a *MTContext* e nem para *Ad Hoc*, pois todas as apps e abordagens foram utilizadas por todos os participantes. Além disso, na primeira sessão, não houve um treinamento de como testar uma AMSC. Os testadores utilizaram seus conhecimentos e experiências para testar as apps.

4.6. Resultados do Estudo de Viabilidade

Os resultados do estudo de viabilidade serão apresentados nas seções seguintes de acordo com a análise quantitativa e a análise qualitativa executadas neste estudo.

4.6.1. Análise Quantitativa

Após as listas de casos de teste gerados terem sido integradas a uma só lista, foi realizada a análise de dados. Essa análise foi feita em ralação à eficácia (número de casos de teste válidos gerados), cobertura (quantidade de contextos alcançados pelos casos de teste e quantidade de variações cobertas) e eficiência (número de casos de teste gerados dividido pelo tempo para a geração, número de contextos divididos pelo tempo para geração e número de variações cobertas dividida pelo tempo para a geração). A Tabela 4 apresenta os resultados individuais obtidos por participante em relação às métricas analisadas no estudo.

As análises quantitativas foram divididas nas variáveis Casos de Teste, Percentual de Testes Válidos, Percentual de Testes Inválidos, Percentual de Contextos, Percentual de Variações, Tempo, Testes Válidos dividido pelo Tempo, Contextos dividido pelo Tempo e Variações dividida pelo Tempo serão apresentadas nas próximas seções. As demais variáveis que não foram analisadas tratam-se de valores absolutos e por isso não foram levadas em consideração para uma possível comparação entre as abordagens ou aplicações. A análise de dados será descrita de acordo com cada variável, sendo primeiro analisada a média. Em seguida serão discutidas as análises estatísticas onde serão mostrados se a população é normal ou não e se a diferença entre as abordagens ou aplicações é significante estatisticamente.

Tabela 4 Resultado Individual de Cada Participante

Abordagem	Projeto	Participante	Casos de Teste	Testes Válidos	%Testes Válidos	Testes Inválidos	%Testes Inválidos	Total de Contextos	% Contextos	Total de Variações	% Variações	Tempo	Testes Válidos / Tempo	Contextos / Tempo	Variações / Tempo
		P1	16	13	81,25	3	18,75	5	27,78	13	86,67	29	0,45	0,17	0,45
		P2	11	11	100,00	0	0,00	16	88,89	11	73,33	34	0,32	0,47	0,32
	APP 1	P3	15	13	86,67	2	13,33	5	27,78	13	86,67	38	0,34	0,13	0,34
		P4	18	13	72,22	5	27,78	6	33,33	13	86,67	22	0,59	0,27	0,59
Ad Hoc		P5	21	13	61,90	8	38,10	7	38,89	13	86,67	47	0,28	0,15	0,28
Adrioc		P6	12	10	83,33	2	16,67	5	50,00	10	100,00	40	0,25	0,13	0,25
	A DD 0	P7	12	10	83,33	2	16,67	8	80,00	10	100,00	35	0,29	0,23	0,29
	APP 2	P8	11	10	90,91	1	9,09	5	50,00	10	100,00	25	0,40	0,20	0,40
		P9	19	13	68,42	4	21,05	9	90,00	10	100,00	33	0,39	0,27	0,30
		P10	26	16	61,54	15	57,69	8	80,00	10	100,00	43	0,37	0,19	0,23
		P1	10	10	100,00	0	0,00	10	100,00	10	100,00	25	0,40	0,40	0,40
		P2	18	10	55,56	8	44,44	10	100,00	10	100,00	20	0,50	0,50	0,50
	APP 2	P3	9	9	100,00	0	0,00	9	90,00	9	90,00	48	0,19	0,19	0,19
		P4	14	10	71,43	4	28,57	10	100,00	10	100,00	27	0,37	0,37	0,37
MTContext		P5	11	9	81,82	2	18,18	9	90,00	10	100,00	43	0,21	0,21	0,23
MTContext -		P6	16	16	100,00	0	0,00	16	88,89	14	93,33	14	1,14	1,14	1,00
		P7	18	14	77,78	4	22,22	14	77,78	12	80,00	38	0,37	0,37	0,32
	APP 1	P8	15	15	100,00	0	0,00	15	83,33	13	86,67	22	0,68	0,68	0,59
		P9	18	14	77,78	4	22,22	14	77,78	12	80,00	33	0,42	0,42	0,36
		P10	33	16	48,48	17	51,52	16	88,89	14	93,33	48	0,33	0,33	0,29

A análise estatística do estudo foi feita seguindo os seguintes passos para cada variável:

- Primeiramente foi realizada a Análise de Normalidade (Shapiro-Wilk Test) com o objetivo de avaliar se a distribuição obtida era normal ou não.
- Para as distribuições normais, aplicou-se o teste paramétrico (teste *t*) com nível de significância de 95%.
- Para variáveis não normais, aplicou-se o teste não paramétrico (*mann-whitney*) com nível de significância de 95%.

A Tabela 5 apresenta os resultados dos testes estatísticos contendo os dados sobre o teste de normalidade, a média por aplicação e por abordagem avaliada e se houve ou não diferença significante entre as abordagens *MTContext* e Ad Hoc ou as aplicações APP1 e APP2. Esses dados estão dispostos de acordo com as variáveis.

Tabela 5 Resultados do Teste Estatístico.

Variável	Fator	Valor	Média	Normal	p-value	Significante?	
Casos de	A b o velo e o o	Ad Hoc	16,1	SIM	0.9704	NÃO	
	Abordagem	MTContext	16,2	SIM	0.9704	NAO	
Teste	Anligação	App 1	18,1	SIM	0.1355	NÃO	
	Aplicação	App 2	14,2	SIM	0.1333	INAU	
	Abordagem	Ad Hoc	78,95%	SIM	0.7511	NÃO	
% Testes	Abordagem	MTContext	81,29%	SIM	0.7511	INAC	
Válidos	Aplicação	App 1	80,60%	SIM	0.8945	NÃO	
	Aplicação	App 2	79,63%	SIM	0.0945	INAC	
	Abordagem	Ad Hoc	21,91%	SIM	0.6905	NÃO	
% Testes	Abordagem	MTContext	18,71%	SIM	0.0905	INAC	
Inválidos	Aplicação	App 1	19,39%	SIM	0.8187	NÃO	
	Aplicação	App 2	21,23%	SIM	0.0107	INAC	
	Abordagem	Ad Hoc	56,67%	SIM	0.0011	SIM	
%	Abordagem	MTContext	89,67%	SIM	0.0011	Silvi	
Contextos	Aplicação	App 1	63,33%	NÃO	0.02574	SIM	
		App 2	83,00%	SIM	0.02374	Silvi	
	Abordagem	Ad Hoc	92,00%	NÃO	0.9329	NÃO	
% Variações	Abordagem	MTContext	92,33%	SIM	0.9329	INAC	
/o variações	Anlicação	App 1	85,33%	SIM	0.00034	SIM	
	Aplicação	App 2	99,00%	NÃO	0.00034		
	Abordogom	Ad Hoc	34,6	SIM	0.5459	NÃO	
Tompo	Abordagem	MTContext	31,8	SIM	0.5459	INAC	
Tempo	Anline a ~ a	App 1	32,5	SIM	0.7000	NÃO	
	Aplicação	App 2	33,9	SIM	0.7636	NAO	
Testes	Abordogo	Ad Hoc	0,36	SIM	0.2204	NÃO	
	Abordagem	MTContext	0,46	SIM	0.3294	INAU	
Válidos / Tempo	Anligação	App 1	0,49	NÃO	0.4000.4	NÃO	
	Aplicação	App 2	0,34	SIM	0.18684	INAU	

	A la a valla ava va	Ad Hoc	0,22	SIM	0.0400	CIM
Contextos / Tempo	Abordagem	MTContext	0,46	SIM	0.0189	SIM
	Aplicação	App 1	0,41	SIM	0.1808	NÃO
	Aplicação	App 2	0,26	SIM	0.1606	INAC
	Abordagem	Ad Hoc	0,34	SIM	0.3402	NÃO
Variações /		MTContext	0,42	SIM	0.3402	INAC
Tempo	Anlicação	App 1	0,45	NÃO	0.11184	NÃO
	Aplicação	App 2	0,32	SIM	0.11104	INAU

4.6.1.1. Casos de teste

Analisando a variável Casos de teste, percebe-se que não há muita diferença entre as médias das abordagens *MTContext* (16,2) e Ad Hoc (16,1) e o mesmo acontece entre as aplicações APP1(18,1) e APP 2 (14,2) em que há uma diferença um pouco maior, porém não significante.

No teste de normalidade aplicado nas abordagens e nas aplicações, ficou constatado que todas as duas abordagens e as duas aplicações são normais e por isso foi aplicado o teste paramétrico em que se percebeu que realmente não há significância estatística tanto entre as abordagens *MTContext* e Ad Hoc (com *p-value* = 0.9704) quanto para as aplicações (com *p-value* = 0.1355).

4.6.1.2. Percentual de Casos de Testes válidos

As médias da variável Percentual de Casos de teste válidos mostram que há uma diferença entre a *MTContext* (81,29%) e Ad Hoc (78,95%) a favor da *MTContext* e também entre a APP1 (80,60%) e APP2 (79,63%), porém esta diferença também não é significante.

Isso novamente se reflete nos testes estatísticos que mostram que não há diferença significante entre as abordagens com p-value = 0.7511 e também não houve influência das aplicações com p-value = 0.8945 (ver A análise estatística do estudo foi feita seguindo os seguintes passos para cada variável:

- Primeiramente foi realizada a Análise de Normalidade (Shapiro-Wilk Test) com o objetivo de avaliar se a distribuição obtida era normal ou não.
- Para as distribuições normais, aplicou-se o teste paramétrico (teste t) com nível de significância de 95%.
- Para variáveis não normais, aplicou-se o teste não paramétrico (mann-whitney) com nível de significância de 95%.

A Tabela 5 apresenta os resultados dos testes estatísticos contendo os dados sobre o teste de normalidade, a média por aplicação e por abordagem avaliada e se houve

ou não diferença significante entre as abordagens MTContext e Ad Hoc ou as aplicações APP1 e APP2. Esses dados estão dispostos de acordo com as variáveis.

Tabela 5).

4.6.1.3. Percentual de Casos de Teste Inválidos

Olhando para as médias do percentual de casos de teste inválidos, é possível notar que também não houve diferença significante tanto para *MTContext* (18,71%) e Ad Hoc (21,91%), quanto para APP1 (19,39%) e APP2 (21,23%), apesar de a média da *MTContext* ter sido melhor que a abordagem Ad Hoc.

Assim como na variável anterior, a população das abordagens e aplicações também são normais e o teste estatístico paramétrico confirmou que não há diferença significante quanto ao percentual de casos de teste inválidos com o *p-value* = 0.6905 para as abordagens e o *p-value* = 0.8187 para as aplicações.

4.6.1.4. Percentual de Contextos Cobertos

Com as médias de *MTContext* (89,67%) e Ad Hoc (56,67%) é possível notar uma ampla diferença entre as abordagens, assim como nas médias de APP1 (63,33%) e APP 2 (83%) também é possível notar uma certa diferença.

Com exceção da APP 1 que sua população não é normal, a população das abordagens e da APP 2 com relação ao percentual de contextos cobertos é normal. Logo, para as abordagens foi realizado um teste paramétrico e para as aplicações foi realizado um teste não paramétrico. O resultado do teste nos diz que há sim uma diferença estatística significante entre as abordagens com *p-value* = 0.0011, o que é positivo, pois mesmo com resultados semelhantes em número de casos de teste, percentual de casos de teste válidos e percentual de casos de teste, isso seria indício que a abordagem faz o que se propõe a fazer, que é a cobertura mais ampla de contextos durante a geração de casos de teste. Analisando as aplicações, também houve diferença significativa com *p-value* = 0.02574, o que indica que houve diferença entre as aplicações, ou seja, uma aplicação foi mais fácil de se cobrir os contextos. Porém, mesmo existindo essa diferença entre as aplicações, isso não seria um problema em relação à avaliação das abordagens, pois todos os participantes utilizaram as duas aplicações e as duas abordagens, logo este viés pode ser descartado.

4.6.1.5. Percentual de Variações de Contexto

Variações de contexto são todas as possibilidades disponíveis pelos elementos de contexto no dispositivo móvel. Ao analisar esta variável, percebeu-se pelas médias da *MTContext* (92,33%) e Ad Hoc (92%) que não há grande diferença entre as abordagens. Isso não é ruim, pois as médias são altas, ou seja, a *MTContext* consegue manter um padrão alto de cobertura das variações de contexto presentes nas aplicações móveis. Já analisando a APP 1 (85,33%) e APP2 (99,00%), percebe-se que existe uma ampla diferença entre as aplicações. Assim como na variável anterior, esta diferença já estava prevista e logo, não é problema ou influência das aplicações na cobertura de variações de contexto entre as abordagens.

Teste não paramétricos foram realizados nesta variável, pois pelo menos uma das abordagens e uma das aplicações não possuíam população normal.

Como resultado, ficou comprovado que estatisticamente não há diferença entre as abordagens com *p-value* = 0.9329 e há diferença entre as aplicações com o *p-value* = 0.00034, indicando que uma das aplicações favorecia a identificação de variações de elementos.

4.6.1.6. Tempo

Analisando a variável tempo, esperava-se que a abordagem MTContext operasse com um tempo maior devido as informações a serem inseridas na ferramenta e no fim. No entanto, não houve uma diferença tão significativa entre as abordagens, sendo que a *MTContext* ainda foi realizada em menos tempo (31,8 minutos) que a forma Ad Hoc (34,6 minutos). Entre as aplicações houve uma diferença menor: APP1 (32,5 minutos) e APP2 (33,9 minutos).

Nos testes estatístico, como todas as variáveis foram consideradas normais, foram realizados somente testes paramétricos para avaliar se houve ou não significância. Com *p-value* = 0.5459, percebeu-se que não houve diferença estatística entre as abordagens e nem entre as aplicações, com *p-value* = 0.7636. Isso bom, pois para abordagens de teste sistemáticas, espera-se que o tempo para a execução seja alto e não foi o que aconteceu neste caso.

4.6.1.7. Testes Válidos Divididos pelo Tempo

Outra variável que não apresentou grande diferença entre as abordagens e as aplicações foi a de teste válidos divido pelo tempo. Mesmo sem grandes diferenças, a média

da *MTContext* (0,46) se mostra maior que a média da abordagem Ad Hoc (0,36). As aplicações ficaram com médias de APP1 (0,49) e APP2 (0,34).

Nesta variável o resultado apresenta as abordagens com *p-value* = 0.3294 indicando que não há diferença estatística entre elas e apresenta as aplicações com *p-value* = 0.18684, o que representa que as aplicações também não tiveram influência nesta variável.

4.6.1.8. Contextos Cobertos Divididos pelo Tempo

Nesta variável, contextos cobertos dividido pelo tempo, é notável a diferença entre a abordagem *MTContext* (0,46) e a abordagem Ad Hoc (0,22). Isso indica que a *MTContext* consegue cobrir mais contextos em menos tempo que a abordagem Ad hoc. Outro resultado importante é que as aplicações não aparecem com médias tão diferentes com a APP1 (0,41) e a APP2 (0.26).

Estatisticamente existe diferença significativa entre as abordagens com p-value = 0.0189 e não há diferença significante entre as aplicações, apresentando *p-value* = 0.1808, ou seja, não houve influência das aplicações nesta variável.

4.6.1.9. Variações de Contextos Divididas pelo Tempo

Finalmente, ao analisar esta variável, percebe-se também que houve diferença tanto para as abordagens quanto para as aplicações. As médias foram *MTContext* (0, 42) e Ad Hoc (0,34). O mesmo se dá para as aplicações, com médias de APP1 (0,45) e APP2 (0,32).

Com testes estatísticos, percebeu-se que estatisticamente não há diferença entre as abordagens e entre as aplicações. Respectivamente, elas possuem *p-value* = 0.3402 e *p-value* = 0.11184. Isso é um bom indício, pois este tempo para cobertura das variações pode ser considerado como um tempo mínimo razoável para a execução da abordagem. E mesmo sem diferença estatística, ainda assim a MTContext conseguiu ter uma média maior que a concorrente, sem ter influência das aplicações.

4.6.2. Análise Qualitativa

A análise qualitativa foi realizada por meio de um questionário online que continham afirmações sobre a usabilidade da ferramenta e comparações sobre as abordagens. Todos participantes responderam este questionário no último dia após a finalização dos casos de

teste criados na *MTContext*. As questões contidas nesse formulário, bem como as respostas dos participantes são mostradas na Tabela 6.

O questionário apresentado está exposto da seguinte forma: para cada questão há uma pontuação de 1 a 5, em que 1 significa discordo totalmente, 2 significa que o participante discorda parcialmente da afirmação, 3 que o participante não tem opinião formada sobre o assunto, 4 que o participante concorda parcialmente com a afirmação feita e 5 que o participante concorda totalmente com a afirmação contida no questionário. Além disso, para cada questão, o participante poderia inserir opiniões, dúvidas ou qualquer comentário a respeito da ferramenta. Por fim, o resultado foi agrupado em positivo e negativo de forma que se a resposta fosse 1 (Discordo Totalmente) ou 2 (Discordo Parcialmente), pertenceria ao grupo Negativo. Caso a resposta fosse 4 (Concordo Parcialmente) ou 5 (Concordo Totalmente), pertenceria ao grupo positivo. As respostas 3 (Não tenho opinião formada) são contabilizadas, mas não comentadas neste trabalho.

Tabela 6 Resultados da Análise Qualitativa.

Questões		Participantes										
		P2	P3	P4	P5	P6	P7	P8	P9	P10		
1. Os "Tipos de Elementos" e as "Variações de Contexto" estão descritas de forma clara e objetiva na ferramenta.	4	5	5	5	5	4	5	3	3	4	0%	80%
2. A navegação dentro da ferramenta é feita de forma fácil e intuitiva.	5	4	5	5	5	5	5	3	3	4	0%	80%
3. A tela de resumo "Test Context - Resume" contém todas as informações cadastradas até ali de forma ordenada e fácil para visualização do usuário.	5	5	5	5	5	5	5	5	4	5	0%	100%
4. As sequências de teste geradas pela ferramenta estão exibidas de forma organizada e de fácil manuseio para cadastro dos campos "Behavior" e "Screenshots".	4	5	5	5	5	5	5	5	4	4	0%	100%
5. A "Dashboard" da ferramenta possui as informações necessárias para um caso de testes de aplicações sensíveis ao contexto. Logo, não há a necessidade de exclusão ou inclusão de alguma outra informação na "Dashboard".	4	4	5	5	5	5	4	4	2	5	10%	90%
6. Todas as informações contidas na "folha de impressão" da "Dashboard" são importantes. Sendo assim, não há a necessidade de exclusão ou inclusão de alguma outra informação na folha de impressão.	4	5	5	5	5	5	4	4	3	5	0%	90%
7. É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem MTContext que normalmente não é possível encontrar apenas com a abordagem Adhoc.	3	4	5	5	4	4	3	5	2	3	10%	60%
8. É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem Adhoc que é não possível encontrar apenas com a abordagem MTContext.	3	2	1	2	2	2	3	3	3	2	60%	0%
9. Você utilizaria futuramente a abordagem MTContext para geração de casos de testes para aplicações móveis sensíveis ao contexto.	4	5	5	5	5	5	5	5	2	4	10%	90%

10. A abordagem MTContext gera casos de teste demais e muitos desses casos de teste são desnecessários para aplicações móveis sensíveis ao contexto.	2	2	1	2	2	1	2	2	3	4	80%	10%
Legenda:												
Med: Mediana; (1) Discordo Totalmente; (2) Discordo; (3) Não tenho opinião formada; (4) Concordo; (5)												
Concordo Totalmente.												

Como pode ser visto na Tabela 6, para a afirmação 1 sobre a clareza do significado dos itens na ferramenta tem 80% de respostas positivas e nenhuma resposta negativa, o que significa que a maioria concorda plenamente que os itens Elemento e Tipos de elementos estão bem definidos na ferramenta. Isso é um bom indício, pois a falta de entendimento por parte do usuário ao utilizar a ferramenta não será problema neste caso. Alguns comentários foram coletados, tais como "Deveria ser melhor explicitado os limites de start e end. Fiquei confuso onde colocaria os números abaixo ou acima, se em start ou end". Para uma evolução da ferramenta, estes campos dentro da ferramenta estarão de forma mais clara aos usuários.

Na segunda afirmação, é dito que "A navegação dentro da ferramenta é feita de forma fácil e intuitiva". As respostas são 80% positivas e novamente não tem nenhuma resposta negativa a esta afirmação. Ou seja, a maioria concorda plenamente que a ferramenta possui uma navegação fácil, e isso não pode então ser considerado um problema para a ferramenta. Alguns comentários foram coletados, tais como "Falta opções de voltar em algumas páginas e se for preciso editar um teste deveria poder "apagar" e não só adicionar um novo elemento" e "Existem algumas falhas na usabilidade da aplicação, me confundi em algumas ocasiões e em algumas funcionalidades, como por exemplo: não consegui visualizar a quantidade de casos de teste gerados pela ferramenta na Dashboard". Como a ferramenta está em fase BETA, estes comentários de usabilidade serão levados em conta para uma evolução da ferramenta.

Na terceira afirmação, correspondente à tela de resumo que deve conter todas as informações já cadastradas até aquele ponto, foi afirmado que "A tela de resumo "Test Context - Resume" contém todas as informações cadastradas até ali de forma ordenada e fácil para visualização do usuário". O índice de respostas positivas é de 100%. Logo, todos concordam que todas as informações necessárias estão dispostas nesta tela. Isso é bom, pois a falta de informações pode ser considerada um problema para abordagens e ferramentas e este não foi o caso neste estudo.

Na quarta afirmação, "As sequências de teste geradas pela ferramenta estão exibidas de forma organizada e de fácil manuseio para cadastro dos campos "Behavior" e

"Screenshots", novamente o índice de respostas positivas é de 100%, mostrando que a todos os participantes também concordam que as sequências de teste geradas estão apresentadas em uma forma de fácil compreensão e os campos para cadastro de comportamentos e *screenshots* também estão fáceis e intuitivos.

Na quinta afirmação, ainda sobre a disposição das informações necessárias para a ferramenta, "A Dashboard da ferramenta possui as informações necessárias para um caso de testes de aplicações sensíveis ao contexto. Logo, não há a necessidade de exclusão ou inclusão de alguma outra informação na Dashboard", foi obtido 90% de respostas positivas e apenas 10% de respostas negativas, mostrando que a maioria dos participantes concorda plenamente que a Dashboard da ferramenta possui todas as informações necessárias para um caso de teste. Esses 10% negativos podem ser explicados com o comentário: "Na Dashboard da ferramenta poderia ser exibido o número de casos de teste gerados pela ferramenta". Esta sugestão servirá também como uma possível evolução para a ferramenta.

Na sexta afirmação, e última sobre a disposição de informações necessárias na ferramenta, "Todas as informações contidas na "folha de impressão" da "Dashboard" são importantes. Sendo assim, não há a necessidade de exclusão ou inclusão de alguma outra informação na folha de impressão", foram recebidas 90% de respostas positivas e nenhuma resposta negativa, o que mostra que a maioria dos participantes concorda plenamente que todos os dados contidos na folha de impressão são importantes para o testador. Alguns comentários foram coletados, tais como "Na folha de impressão da ferramenta poderia ser exibido o número de casos de teste gerados pela ferramenta". Este comentário é muito parecido com o comentário feito na afirmação anterior e assim como solução dita anteriormente, esta informação sugerida por um dos participantes pode sim ser inserida em uma possível evolução da ferramenta.

Na sétima afirmação, referente às abordagens, "É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem *MTContext* que normalmente não é possível encontrar apenas com a abordagem Adhoc", as respostas ficaram divididas em 60% positivas e 10% negativas, o que mostra que a maioria dos participantes entende que há uma vantagem na geração de casos de teste ao utilizar a MTContext e que essa vantagem é a de poder gerar casos de tese que normalmente não seriam gerados com uma abordagem Ad Hoc. Não foi deixado nenhum comentário sobre o motivo do descontentamento ou uma sugestão de melhoria por parte dos 10% de respostas negativas. Porem um comentário foi coleado, "Acredito que seja porque o *MTContext* já possua parâmetros pré-programados que ajudam na definição de casos de teste". Este participante comenta sobre como ele entendeu o funcionamento da ferramenta e como ela

pode ser capaz de criar casos de teste que porventura não tenham sido criados em uma outra abordagem como a Ad Hoc.

Na oitava afirmação, "É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem Adhoc que não é possível encontrar apenas com a abordagem MTContext.", foram coletadas 10% de respostas positivas e 60% de respostas negativas. No entanto, pode-se perceber que nesta afirmação, as respostas negativas indicam que Ad Hoc não seria tão abrangente quando MTContext, ou seja, eles não acreditam que a abordagem Ad Hoc possa gerar casos de teste que não poderiam ser gerados apenas com a *MTContext*. Essas seriam, ao final, respostas favoráveis à MTContext. Apenas 10% das respostas (as positivas) foram desfavoráveis. Novamente, nenhum comentário de sugestão de melhoria ou reclamação foi identificado. Porém, houve um comentário: "Acredito que MTControl consegue abranger até mais opções em relação ao Adhoc".

Na nona questão, referente somente à abordagem *MTContext*, "Você utilizaria futuramente a abordagem MTContext para geração de casos de testes para aplicações móveis sensíveis ao contexto", as respostas resultaram em 90% positivas e 10 negativas, mostrando que a maioria dos usuários usaria a abordagem novamente para a geração de casos de teste para aplicações móveis sensíveis ao contexto. Isto serve de incentivo para a continuação deste trabalho e de mais estudos na área de teste para aplicações móveis sensíveis ao contexto. Nenhum comentário apontando o descontentamento por parte dos 10% foi identificado. O único comentário feito foi, "Acredito que esta abordagem otimiza bastante o processo comparado ao Adhoc".

Na décima e última questão, ainda somente sobre a abordagem *MTContext* "A abordagem *MTContext* gera casos de teste demais e muitos desses casos de teste são desnecessários para aplicações móveis sensíveis ao contexto", o resultado das respostas é de 80% negativas e 10% positivas. No entanto, assim como a afirmação 8, pode-se perceber que nesta afirmação, as respostas negativas são favoráveis à *MTContext*, pois indicam que os casos de teste gerados por MTContext não são exagerados ou desnecessários. Isto é um bom indício, pois um dos principais problemas na geração de casos de teste para aplicações sensíveis ao contexto é uma possível geração exponencial de contextos para essas aplicações, o que torna o teste dessas aplicações ainda mais difícil e custoso devido ao grande número de casos de teste para essas aplicações. Alguns comentários foram coletados, tais como "Depende muito da aplicação a ser analisada, não vale para todas". Este participante comenta que ele acredita que pode acontecer que dependendo da aplicação, pode ser que a *MTContext* gere casos de teste desnecessários.

4.7. Considerações Finais

Nesta seção, foi apresentado um estudo de viabilidade realizado com o objetivo de comparar a abordagem *MTContext* com uma abordagem Ad Hoc a fim de se obter indícios sobre a possibilidade de seu uso para geração de casos de teste para aplicações móveis sensíveis ao contexto. Após analisar os resultados com as médias por variáveis avaliadas, observou-se que a abordagem *MTContext* foi melhor em algumas variáveis avaliadas e em outras variáveis, pode-se dizer que ela obteve o mesmo desempenho que a abordagem Ad Hoc, pois a diferença entre elas não foi estatisticamente significante. Isto indica que esta abordagem pode sim ser viável para ser utilizada no apoio a geração de casos de teste para aplicações móveis sensíveis ao contexto.

Nas variáveis Casos de Teste, Percentual de Testes Válidos, Percentual de Testes Inválidos, Tempo para a geração dos testes e Testes Válidos dividido pelo **Tempo** a MTContext obteve um desempenho estatisticamente não significante em relação a abordagem Ad Hoc e em nenhuma dessas variáveis, houve influência das aplicações utilizadas. Já na variável Percentual de Variações, além de existir diferença entre as abordagens, houve diferença significante entre as aplicações. Isso indica que a aplicação pode ter influenciado nesta variável. Além disso, na variável Contexto que indica a quantidade de contextos cobertos pelos casos de teste houve diferença entre as abordagens e isso é bom para a MTContext, porém também houve diferença entre as aplicações e isso é ruim pois é um indício que houve influência das aplicações nesta variável. Contudo, na variável Percentual de Contextos dividido pelo Tempo que é a variável que faz a relação entre a quantidade de contextos pelo tempo de geração de casos de teste, houve diferença significante entre as abordagens e isso é muito bom para a MTContext, pois esta é a principal variável para a avaliação do que a abordagem tem como objetivo. Além disso, não houve diferença entre as aplicações o que indica que nenhuma das aplicações teve influência nesta variável.

Com a análise qualitativa foi possível observar possíveis melhorias na ferramenta e também na abordagem, como por exemplo a adição de informações de casos de teste gerados pela ferramenta, mesmo que esses casos de teste sejam inválidos. A análise qualitativa também possibilitou ter um feedback importante dos usuários (testadores) da ferramenta, pois ela foi destinada para os mesmos. Além disso, uma das principais informações alcançadas por meio do questionário foi que a maioria dos usuários voltaria a usar a ferramenta outra vez e que ela pode sim facilitar o processo de geração de casos de teste para aplicações móveis sensíveis ao contexto.

Sendo assim, é possível afirmar que a abordagem *MTContext* é viável para ser utilizada no que diz respeito a geração de casos de teste para aplicações móveis sensíveis ao contexto, respondendo à hipótese definida no início deste trabalho.

CAPÍTULO 5 - CONCLUSÃO

Neste capítulo serão apresentadas as conclusões desta dissertação, ressaltando suas contribuições e trabalhos futuros que fornecem a direção para que seja dada continuidade a esta pesquisa, além de suas limitações.

5.1. Considerações Finais

Este trabalho apresentou a criação de uma abordagem para apoio à geração de casos de teste para aplicações móveis sensíveis ao contexto. Este apoio é fornecido por meio de uma ferramenta, *MTContext*, que possibilita a fácil implementação dos paradigmas definidos na abordagem. Neste contexto, o objetivo desta abordagem foi facilitar a geração de contextos para aplicações móveis e casos de testes para esses contextos de forma automatizada e sistemática.

Como contribuição, a abordagem *MTContext* fornece uma forma de extrair informações de contextos partindo de requisitos das aplicações móveis. Essas informações podem ser inseridas no apoio ferramental e a partir dessas informações, irá gerar diferentes contextos e casos de teste de uma forma não exponencial para que a execução desses testes não seja inviável. Além disso, os casos de teste são criados de acordo com elementos nativos dos dispositivos móveis e possuem uma sequência lógica baseada em máquinas de estado finita, o que torna sua execução possível tanto manualmente quanto automatizada.

5.2 Contribuições

As principais contribuições oferecidas por esta pesquisa à comunidade de teste de software são:

- Uma abordagem para apoiar a geração de casos de teste em aplicações móveis sensíveis ao contexto.
- Uma ferramenta que implementa um apoio à geração de casos de teste em aplicações móveis sensíveis ao contexto.
- Uma Base de conhecimento sobre as características relevantes de teste em aplicações móveis sensíveis ao contexto que poderá ajudar pesquisadores que desejam trabalhar com essas áreas.

5.3 Limitações

- O estudo de avaliação da abordagem não é conclusivo, pois foi aplicado em contexto específico que não permite a expansão de seus resultados para outros cenários. Novos estudos em outros cenários com diferentes aplicações são necessários.
- A abordagem necessita de constantes atualizações a cada característica nova de sensibilidade ao contexto encontrada e para quais dispositivos elas estão disponíveis.
- A ferramenta está disponível apenas no idioma inglês e sua utilização pode ser difícil para usuário que não detém domínio sobre o idioma.

5.4 Trabalhos futuros

Apesar do estudo de avaliação não ser conclusivo, os dados obtidos nos estudos indicam que a abordagem é viável para o que se propôs realizar. Assim, como próximos passos estão previstos:

- A execução da metodologia proposta em (SHULL et al., 2001) para ser seguida na avaliação da MTContext, visando a longo prazo sua transferência para ambiente industrial. Estes novos estudos contribuiriam para a evolução e o amadurecimento da abordagem, propiciando melhorias na abordagem.
- Integrar a MTContext a um outro apoio ferramental que possibilite a execução automatizada dos casos de teste gerados pela MTContext. Essa integração já pode ser percebida em (COSTA et al., 2015), onde os autores criaram scripts para a execução de teste aplicações móveis sensíveis ao contexto por meio de elementos nativos dos dispositivos comparando suas telas (screenshots), assim como a abordagem MTContext gera seus casos de teste.
- Aprimorar o algoritmo para a geração dos contextos de forma que mais contextos possam ser gerados e menos casos de teste sejam necessários.
- Inserir outras características de ubiquidade no apoio ferramental para que seja possível não só os testes para aplicações móveis sensíveis ao contexto e sim para aplicações móveis ubíquas de acordo com suas características definidas por Spínola (2010).

REFERÊNCIAS BIBLIOGRÁFICAS

- Abowd, G. D.; Dey, A. K.; Brown, P. J.; Davies, N.; Smith, M.; Steggles, P.; "Towards a Better Understanding of Context and Context-Awareness". In Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing (HUC'99), Hans-Werner Gellersen (Ed.). Springer-Verlag, London, UK, pp. 304-307, 1999
- Amalfitano, D.; Fasolino, A.R.; Tramontana, P.; Amatucci, N., "Considering Context Events in Event-Based Testing of Mobile Applications," Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on, vol., no. pp.126,133, 18-22 March 2013
- Araújo, R. B. DE. "Computação Ubíqua: Princípios, Tecnologias e Desafios". XXI Simpósio Brasileiro de Redes de Computadores. Anais...2003
- Barbosa, E. F., Maldonado, J. C., Vincenzi, A. M. R., Delamaro, M. E., Souza, S. R. S., & Jino, M. (2000). "Introdução ao teste de software". Minicurso apresentado no XIV Simpósio Brasileiro de Engenharia de Software (SBES 2000).
- Basili, V.R.; Caldiera, G.; Rombach, H.D. (1994) "Goal Question Metric Approach". Il Encyclopedia of Software Engineering, John Wiley & Sons, Inc, pp. 528-532.
- Brito, J.; C.; "Testcheck: Uma Abordagem Baseada Em Checklist Para Inspecionar Artefatos De Teste De Software"; Dissertação de Mestrado; Universidade Federal do Amazonas UFAM- 2012
- Cano, J. C., Ferrández-Bell, D., & Manzoni, P.; "Evaluating bluetooth performance as the support for context-aware applications." Telecommunication Systems, 28(3-4), 333-347. (2005).
- Chang, T. H.; Yeh, T.; and Miller, R.; "GUI Testing using Computer Vision," Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, New York, USA, pp. 1535-1544. (2010)
- Chen, G. and Kotz, D.; "A Survey of Context-Aware Mobile Computing Research," Hanover, NH, USA, Tech. Rep., 2000
- Costa, E. S., Reis, R. D. A. C., & Dias-Neto, A. C. (2015).; "Extension of Sikuli Tool to Support Automated Tests to Windows Phone Context-Aware Applications." UBICOMM 2015

- Craig, R.D., Jaskiel, S. P., "Systematic Software Testing", Artech House Publishers, Boston, 2002
- Dantas V. L. L.; "Requisitos para testes de aplicações móveis". Dissertação de mestrado, Programa de Pós-graduação em Ciência da Computação (MDCC) da Universidade Federal do Ceará (UFC). 2009
- Delamaro, M.E., Maldonado, J.C., Jino, M., "Introdução ao Teste de Software", Rio de Janeiro: Elsevier. (2007),
- Dey, A. K.; "Understanding and Using Context." Personal Ubiquitous Comput. 5, 1, 4-7. DOI=http://dx.doi.org/10.1007/s007790170019, (January 2001).
- GAO, J.; Bai, X.; Tsai, W.; Uehara, T.; "Mobile Application Testing: A Tutorial". In: Journal Computer, Vol. 47, pp. 46-55, DOI=10.1109/MC.2013.445, (2014)
- Harrison, R.; Flood, D.; Duce, D.; "Usability of mobile applications: literature review and rationale for a new usability model." Journal of Interaction Science 1 (1), pp. 1-16. 2013
- Hesenius, M.; Griebe, T.; Gries, S.; and Gruhn, Volker.; "Automating UI tests for mobile applications with formal gesture descriptions." In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14). 2014 ACM, New York, NY, USA, 213-222. DOI=http://dx.doi.org/10.1145/2628363.2628391, (2014)
- Huy, N. P.; VanThanh. D.; "Evaluation of mobile app paradigms". Proceedings of International Conference on Advances in Mobile Computing & Multimedia (MoMM '12), Ismail Khalil (Ed.). 2012. Pp. 25-30. DOI=10.1145/2428955.2428968. (2012)
- Kjeldskov, J.; Paay, J.; "A longitudinal review of Mobile HCI research methods." In: International Conference on Human-computer interaction with mobile devices and services (MobileHCI'12), 2012, pp. 69-78. DOI=10.1145/2371574.2371586 (2012)
- Kirubakaran, B.; and Karthikeyani, V.; "Mobile application testing Challenges and solution approach through automation", International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME), pp. 79-84, 2013, doi: 10.1109/ICPRIME.2013.6496451, (2013)
- Kitchenham, B.; Charters, S. (2007) "Guidelines for Performing Systematic Literature Reviews in Software Engineering". Version 2.3. Technical Report, Evidence-Based Software Engineering (EBSE)

- LioNBRIDGE. "Testing Mobile Applications is Different from Testing Traditional Applications".

 Disponível em: http://www.lionbridge.com/lionbridge/en-US/kc/outsourced-testing.htm
 (2006) Acesso em: 06 de outubro de 2014
- Lu, H. "A software testing framework for context-aware applications in pervasive computing." Tese (Doutorado) — Universidade de Hong Kong, 2009.
- Masi, E.; Cantone, G.; Mastrofini, M.; Calavaro, G.; and Subiaco, P.; "Mobile apps development: A framework for technology decision making", in Proceedings of International Conference on Mobile Computing, Applications, and Services (MobiCASE'4), pp. 64–79. (2012)
- Mota, L.; S.; "Uma Abordagem Para Especificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software". Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro. (2013)
- Nascimento, J.; P.; Silva. J.; S.; and Dias-Neto, A.; C.; "MTControol: Ferramenta de Apoio à Recomendação e Controle de Critérios de Teste em Aplicações Móveis", VI Congresso Brasileiro de Software: Teoria e Prática (CBSoft), Sessão de Ferramentas setembro de 2015
- Pinheiro, A. C., Simão, A., & Ambrosio, A. M.; "FSM-Based Test Case Generation Methods Applied to Test the Communication Software on Board the ITASAT University Satellite: A Case Study." Journal of Aerospace Technology and Management, 6(4), 447-461. (2014).
- PNAD, "Acesso à Internet e Posse de Telefone Móvel Celular para Uso Pessoal." IBGE Disponível em:http://www.ibge.gov.br/home/presidencia/noticias/imprensa/ppts/00000012962305122013234016242127.pdf acessado em 2 de fevereiro de 2015 as 14:40 (2013)
- Raymundo, C. R., & Costa, P. D. Uma Plataforma de Testes para Aplicaç oes Sensıveis a Contexto. IV Simpósio Brasileiro de Computação Ubíqua e Pervasiva SBCUP (2012)
- Reis, R. A. C., Fontão, A. D. L., Gomes, L. L., & Dias-Neto, A. C.; "Usability Evaluation Approaches for (Ubiquitous) Mobile Applications: A Systematic Mapping Study." Proceedings of International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies. UBICOMM 2015
- Santos, I.; S.; "Um Ambiente Para Geração De Cenários De Testes Para Linhas De Produto De Software Sensíveis Ao Contexto". Dissertação submetida à Coordenação do

- Programa de Pós-Graduação em Ciência da Computação da Universidade Federal do Ceará UFC. (2013)
- Samsel, C., Beul-Leusmann, S., Wiederhold, M., Krempels, K. H., Ziefle, M., & Jakobs, E. M.; "Cascading Information for Public Transport Assistance." InProceedings of the 10th International Conference on Web Information Systems and Technologies (WEBIST 2014). INSTICC, Barcelona (in press, 2014).
- Satoh. I.; "A testing framework for mobile computing software." IEEE Transactions on Software Engineering, 29(12):1112–1121, Dec 2003.
- Satyanarayanan, M.; "Fundamental Challenges in Mobile Computing", in Proceedings of the fifteenth annual ACM Symposium on Principles of distributed computing (PODC'96), New York, NY, USA: ACM, pp. 1–7, doi: 10.1145/248052, (1996)
- Shull, F.; Carver, J.; Travassos, G.; "An Empirical Methodology for Introducing Software Processes". Proceedings of the Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9), pp. 288-296. (2001)
- Spínola, R. O.; Massollar, J.; "Travassos, G. Checklist to Characterize Ubiquitous Software Projects".; Brazilian Symposium on Software Engineering. Anais...2007
- Spinola, R.O.; Dias-Neto, A.C.; Travassos, G. H. (2008) "Abordagem para Desenvolver Tecnologia de Software com Apoio de Estudos Secundários e Primários". Experimental Software Engineering Latin American Workshop (ESELAW).
- Spínola. R. O., "Apoio A Especificação e Verificação De Requisitos Funcionais De Ubiquidade Em Projetos De Software". Tese de Doutorado. Alberto Luiz Coimbra De Pós-Graduação E Pesquisa De Engenharia (Coppe) Da Universidade Federal Do Rio De Janeiro. 2010
- Spínola, R. O.; Travassos, G. H.; "Towards a framework to characterize ubiquitous software projects. Information and Software Technology". v. 54, p. 759-785. http://dx.doi.org/10.1016/j.infsof.2012.01.009 (2012)
- Tse, T.; Yau, S.; Chan, W.; Lu, H.; and Chen. T.; "Testing context-sensitive middleware-based software applications." In International Computer Software and Applications Conference, pages 458–465, Sept 2004.
- Vieira, V.; Tedesco, P.; Salgado, A. C.; "Modelos e processos para o desenvolvimento de sistemas sensíveis ao contexto". [S.I.], 2009.

- Yau, S.S.; Yu, W.; Karim, F.; "Development of Situation-aware application software for ubiquitous compuing environments" Proceedings of the 26th Annual International Computer Software and Applications Conference. COMPSAC, Page: 233-238. (2002)
- Yu, J., Tappenden, A., Geras, A., Smith, M., & Miller, J. "Agile testing of location based services". In Extreme Programming and Agile Processes in Software Engineering (pp. 239-242). Springer Berlin Heidelberg. 2005
- Wang. Y.; "An FSM model for situation-aware mobile application software systems". InProceedings of the 42nd annual Southeast regional conference (ACM-SE 42). ACM, New York, NY, USA, 52-57. DOI=http://dx.doi.org/10.1145/986537.986551. (2004)
- Wang, Z.; Elbaum, S.; Rosenblum, D. S.; "Automated generation of context-aware tests." In: Proceedings of the 29th international conference on Software Engineering. Washington, DC, USA: IEEE Computer Society, (ICSE '07), p. 406–415.Disponível em: http://dx.doi.org/10.1109/ICSE.2007.18. 2007
- Zhang, D, & Adipat, B.; "Challenges, methodologies, and issues in the usability testing of mobile applications." International Journal of Human-Computer Interaction, 18(3), 293–308. (2005)
- Zeidler, C.; Kittl, C. e Petrovic, O. (2007). "An Integrated Product Development Process for Mobile Software." In: Proceedings of the IX International Conference on the Management of Mobile Business (ICBM), p. 23-23. 2007

APÊNDICES

Apêndice A Termo de Consentimento livre e Esclarecido – TCLE

Experimentação em Teste de Software

Eu declaro ter mais de 18 anos de idade e que concordo em participar em estudos não invasivos conduzidos pelo pesquisador Prof. Dsc. Arilo Claudio Dias Neto e o pesquisador Rodrigo dos Anjos Cruz Reis, como parte das atividades dos cursos de graduação e pós-graduação do IComp da UFAM/AM. Estes estudos visam compreender a viabilidade da aplicação de técnica de geração de testes para aplicações móveis sensíveis ao contexto, tentando entender sob que circunstâncias estas técnicas podem ser efetivamente utilizada.

PROCEDIMENTO

Conceitos sobre qualidade, sensibilidade ao contexto e Teste de Software para aplicações móveis serão apresentados. Eu entendo que serei ensinado como as tecnologias podem ser aplicadas e serei solicitado a aplicá-las no projeto distribuído durante o curso. Neste projeto, técnica de teste usando aplicações móveis sensíveis ao contexto será aplicada por mim visando permitir pensar sobre seu uso e avaliá-lo. Eu entendo que, uma vez o curso terminado, o trabalho que desenvolvi, será estudado visando entender a eficiência e eficácia da técnica que me foi ensinada.

Eu entendo que esta atividade preenche parte dos requisitos do curso e serão avaliados como tal. O pesquisador conduzirá o estudo consistindo da coleta, análise e relato dos dados da atividade de teste. Eu entendo que não tenho obrigação alguma em contribuir com informação sobre meu desempenho na atividade, e que posso solicitar a retirada de meus resultados do experimento a qualquer momento e sem qualquer penalidade ou prejuízo. Eu entendo que não existirá nenhum crédito ou benefício extra por participar deste estudo, e que não haverá qualquer impacto negativo em minha avaliação por não participar do estudo. Eu entendo também que quando os dados forem coletados e analisados, meu nome será removido dos dados e que este não será utilizado em nenhum momento durante a análise ou quando os resultados forem apresentados.

CONFIDENCIALIDADE

Toda informação coletada neste estudo é confidencial, e meu nome não será identificado em momento algum. Da mesma forma, me comprometo a não comunicar os meus resultados enquanto não terminar o estudo, bem como manter sigilo da técnica e documentos apresentados e que fazem parte do experimento.

. BENEFÍCIOS, LIBERDADE DE DESISTÊNCIA

Eu entendo que os benefícios que receberei deste estudo são contemplados em 6 horas complementares ao curso e limitados ao aprendizado do material que é distribuído e ensinado visando atender os requisitos do curso, independente de participar ou não deste estudo, mas que os pesquisadores esperam aprender mais sobre quão eficiente é a utilização de tecnologias de software e os benefícios trazidos por este estudo para o contexto da Engenharia de Software.

Eu entendo que sou livre para realizar perguntas a qualquer momento ou solicitar que qualquer informação relacionada à minha pessoa não seja incluída no estudo. Eu entendo que minha participação no estudo não afetará minha nota final de qualquer forma, e que participo de livre e espontânea vontade com o único intuito de contribuir para o avanço e desenvolvimento de técnicas e processos para a Engenharia de Software.

PESQUISADORES

Rodrigo dos Anjos Cruz Reis Programa Pós-Graduação de em Informática – UFAM Prof. Dsc. Arilo Claudio Dias Neto Programa de Pós-graduação em Informática - UFAM

Apêndice B - Formulário de Caracterização

Este formulário tem por objetivo caracterizar seu nível de conhecimento referente ao conceito de Sensibilidade ao Contexto, experiência em teste de software, experiência em teste de aplicações móveis e teste de aplicações móveis sensíveis ao contexto. As informações resultantes deste formulário serão utilizadas neste experimento para dividir os participantes em dois grupos de modo que ambos possuam igualdade de experiência para que o experimento seja justo. Por favor responda todas as questões de acordo com seus conhecimentos.

Nome_	
E-mail_	

Nível de conhecimento com o conceito de Sensibilidade ao Contexto

- 1. Você possui um Smartphone, tablet ou outro dispositivo móvel? Sim () Não ()
- 2. Das características listadas abaixo, quais fornecem informações para as aplicações que você utiliza de forma que as aplicações se adaptem a elas?
 - GPS () DATA/HORA () BATERIA () ROTAÇÃO DE TELA () CONEXÃO () TEMPERATURA () OUTRO()
- 3. Cite pelo menos 3 (três) desses aplicativos:
- 4. De acordo com seus conhecimento sobre o conceito de Sensibilidade ao contexto marque abaixo as alternativas corretas:
 - () contexto é qualquer informação que pode ser utilizada para caracterizar a situação de uma pessoa, um lugar ou um objeto relevante para a interação entre um usuário e uma aplicação.
 - () contexto é qualquer informação escrita de forma textual que pode ser utilizada para caracterizar a situação de uma pessoa, um lugar ou um objeto relevante para a interação entre um usuário e uma aplicação.
 - () Uma aplicação sensível ao contexto é capaz de identificar ,apenas por sensores, informações que possam modificar seu comportamento e assim a aplicação deve adaptar-se a essas informações.
 - () Sensibilidade ao contexto é quando há alteração/adaptação no comportamento e funcionalidades da aplicação para corresponder ao contexto de uso em causa.

Experiência em teste de Aplicações móveis

- 5. Associe os conceitos de acordo com suas respectivas nomenclaturas.
 - a) valida funções de serviços, APIs web móveis, comportamentos de sistemas externos, interfaces do usuário (UIs) e seus gestos, funções baseadas em localização, perfis de usuários, dados do sistema e dados do usuário.
 - b) visa percorrer os diferentes caminhos possíveis de execução de um código para uma aplicação móvel a fim de atingir uma alta cobertura de código e garantir a qualidade da aplicação sob teste.
 - c) avalia a compatibilidade de aplicações a diferentes navegadores web, plataformas e conectividade com redes de comunicação.
 - d) avalia conteúdo e alertas na interface do usuário, fluxos e cenários de operação do usuário, mídias e apoio a interação por gestos.
 - () a) Teste Funcional e Comportamental. b) Teste de Usabilidade. c) Teste de Compatibilidade e Conectividade. d) Teste Estrutural.
 - () a) Teste de Usabilidade. b) Teste Funcional e Comportamental. c) Teste Estrutural. d) Teste de Compatibilidade e Conectividade.
 - () a) Teste Funcional e Comportamental. b) Teste Estrutural. c) Teste de Compatibilidade e Conectividade. d) Teste de Usabilidade.
 - () a) Teste Funcional e Comportamental. b) Teste de Usabilidade. c) Teste de Compatibilidade e Conectividade. d) Teste Estrutural.

Experiência em teste de aplicações móveis sensíveis ao contexto

Experiência em Teste de Aplicações móveis sensíveis ao contexto onde 1 corresponde ao mais baixo nível e 5 ao mais alto nível de experiência.

6. Experiência em planejar testes

- 1 Nenhuma
- 2 baixa Conheço os conceitos mas nunca pratiquei
- 3 média Conheço os conceitos e já pratiquei em um projeto acadêmico
- 4 alta Conheço os conceitos e já pratiquei em mais de um projeto acadêmico.
- 5 altíssima Pratiquei em um ou mais projetos na indústria.

7 Experiência em projetar casos e procedimentos de testes

- 1 Nenhuma
- 2 baixa Conheço os conceitos mas nunca pratiquei
- 3 média Conheço os conceitos e já pratiquei em um projeto acadêmico
- 4 alta Conheço os conceitos e já pratiquei em mais de um projeto acadêmico.
- 5 altíssima Pratiquei em um ou mais projetos na indústria.

8. experiência com pelo menos uma ferramenta de teste

- 1 Nenhuma
- 2 baixa Conheço os conceitos mas nunca pratiquei
- 3 média Conheço os conceitos e já pratiquei em um projeto acadêmico
- 4 alta Conheço os conceitos e já pratiquei em mais de um projeto acadêmico.
- 5 altíssima Pratiquei em um ou mais projetos na indústria.
- Agradecemos pela sua contribuição neste experimento.

Apêndice C - Requisitos das Aplicações - APP 2 - iFOOD TRUCK

Este documento apresenta um modelo de especificação de requisitos para aplicações móveis sensíveis ao contexto em forma de cenários. Este modelo fornece informações de contexto baseada em teste conforme descrito abaixo.

Imaginemos vários food trucks existentes dentro de uma cidade. A cada dia eles estão em um lugar diferente oferecendo seus produtos. Cada food truck possui um sensor que indica onde esse food truck está no momento e uma câmera que exibe em tempo real a preparação dos pedidos.

DESCRIÇÃO

Foi desenvolvido um aplicativo para o Iphone 5s e 6Plus que utiliza o sistema operacional iOS. Esse aplicativo gerencia os pedidos para esses food trucks. Assim, o usuário pode realizar sua compra de maneira confortável e com antecedência evitando assim a espera pelo produto solicitado. Este aplicativo possui os seguintes requisitos.

Pesquisar Food Trucks

- 1. Ao abrir o aplicativo, deve ser exibida uma lista com os food trucks existentes por ordem:
 - a. Ordem de distância: (menor distância entre ponto de origem do usuário e o ponto do food truck, destino do usuário)
 - b. Ordem de acordo com o clima. Acima de 30 °C a ordem é de preferência para gelaterias, sorveterias ou especializadas em bebidas.

Selecionar Food Truck e o Prato Desejado

- 2. A rotação de tela é permitida somente nesta tela de selecionar pedido, logo esta tela pode ficar tanto na horizontal quanto na vertical
- 3. Após o usuário selecionar o food truck e o prato que deseja, devem estar disponíveis informações de tempo de duração de preparo do prato e tempo para o usuário chegar ao Foodtruck. Após realização de pagamento, o pedido é autorizado e o nome do usuário entra na fila de espera.
- 4. Após ultrapassar o horário de funcionamento, a opção de confirmar o pedido deve ser desabilitada e o motivo deve ser informado.

Acompanhar Pedido

- 5. Após o pedido ser realizado, uma opção de exibição de câmera é oferecida.
 - a. Se a internet for wi fi, ela exibirá ao vivo o preparo do pedido.
 - b. Se a internet for rede de dados, ela apenas exibirá uma tela com a receita e informará a necessidade de wi fi.
 - c. Se a bateria estiver abaixo de 15%, ela apenas exibirá uma tela com a receita e informará que o celular necessita ser carregado.
 - d. Se a bateria estiver acima ou igual a 15% e menor que 40%, ela exibirá uma mensagem que o celular estará prestes a descarregar e que necessita ser carregado para que a exibição do preparo do pedido continue ao vivo.

De acordo com os requisitos especificados acima, imaginemos:

- Cenário 1: O usuário resolve utilizar o aplicativo e pesquisar os food trucks disponíveis.
- Cenário 2: O usuário resolve fazer um pedido para algum food truck selecionado.
- **Cenário 3:** O usuário resolve acompanhar o preparo do seu pedido e vai buscá-lo no respectivo food truck em que foi feito o pedido.

Apêndice D - Requisitos das Aplicações - APP 1 - MINHA BICICLETA

Este documento apresenta um modelo de especificação de requisitos para aplicações móveis sensíveis ao contexto em forma de cenários. Este modelo fornece informações de contexto baseada em teste conforme descrito abaixo.

(Leia toda a descrição antes de criar os casos de teste)

Imaginemos um serviço de bicicletário com pontos espalhados por toda cidade. Devem existir sensores instalados para identificação das bicicletas. Essas bicicletas são alugadas por hora.

DESCRIÇÃO

Foi desenvolvido um aplicativo para controle desse bicicletário. Este aplicativo funciona apenas no dispositivo Moto X que utiliza o Sistema Operacional Android. Este aplicativo possui os seguintes requisitos:

Pesquisar Pontos de Bicicletas

- 1. Ao pesquisar os pontos de bicicletas existentes no campo de busca, uma tela deve mostrar os pontos de bicicletas mais próximos do usuário.
- O aplicativo deve ser capaz de se comunicar com o ponto de bicicleta. Essa comunicação deve ser composta por:
 - a. Autorizar a retirada da bicicleta pelo usuário após pagamento do aluguel da bicicleta se ele estiver a no máximo 3 metros de distância do bicicletário.
 - i. Informar dados de identificação da bicicleta que pode ser retirada, hora da retirada e a quantidade de horas alugadas, ou seja, disponíveis para uso.
 - b. Não autorizar a retirada da bicicleta pelo usuário.
 - i. Informar motivo da não autorização de retirada da bicicleta. (Motivos da não autorização: Estar a mais de 3 metros de um bicicletário e/ou pagamento não autorizado).

Consultar Aplicativo

- 3. Enquanto o usuário estiver com a bicicleta, o aplicativo deve ter as seguintes informações:
 - a. Quando o tempo restante para uso da bicicleta chegar a 10 minutos, o usuário deverá ser informado do tempo restante para uso e uma opção para o aluquel de mais horas
 - Se o usuário ultrapassar o tempo permitido para uso da bicicleta, o sistema de alarme é acionado.
 Para o usuário, deve ser informado o motivo do acionamento do alarme da bicicleta e oferecida a possibilidade de alugar mais horas.
 - c. Înformação da temperatura do ambiente. Se está Frio (abaixo de 21 °C) e Quente (acima de 25°C).
 - d. Estado da Bateria do Celular: Baixa (abaixo 21%) e Alta (entre 90% e 100%).

Devolver Bicicleta

- 4. Ao devolver a bicicleta, o aplicativo deve ser capaz de se comunicar com o ponto de bicicletas. Essa comunicação deve ser composta por:
 - a. Identificar que a bicicleta está corretamente encaixada no bicicletário, travar bicicleta e informar ao usuário a hora da devolução e a quantidade de horas disponíveis para uso.
 - b. Identificar que a bicicleta não está corretamente encaixada e informar ao usuário.
- 5. A rotação de tela só é permitida na tela após pesquisa por pontos de bicicleta. Ou seja, a tela pode ficar em modo Vertical ou Horizontal.

De acordo com os requisitos especificados acima, imaginemos:

Cenário 1: O usuário após pesquisar os pontos de bicicletas mais próximos, resolve alugar uma bicicleta por uma hora.

Cenário 2: O usuário após o aluguel da bicicleta, consulta o aplicativo para obter as informações disponíveis no aplicativo.

Cenário 3: O usuário devolve a bicicleta.

Apêndice E - Roteiro do Participante - 1° Sessão APP1 - Minha Bicicleta

Tarefas

De acordo com o documento de requisitos da Aplicação 1 – Minha Bicicleta, identifique e marque com um
 X, os elementos de contexto existentes em cada cenário.

Cenários				Elementos			
1 <u>Pesquisar Pontos de</u> <u>Bicicletas</u>	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor
2 Consultar Aplicativo	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor
3 <u>Devolver Bicicleta</u>	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor

2. Verifique na descrição da app, quais os requisitos para cada elemento em cada cenário.

Cenário 1 - Pesquisar Pontos de Bicicletas

Elementos	Requisitos do Elemento	Descrição da APP
Localização	(a) (b) (c) (d)	(a)Ao pesquisar os pontos de bicicletas existentes no campo de busca, uma tela deve mostrar os pontos de bicicletas mais
Conexão	(a) (b) (c) (d)	próximos do usuário. O aplicativo deve ser capaz de se comunicar com o ponto de
Tempo	(a) (b) (c) (d)	bicicleta. Essa comunicação deve ser composta por: (b) Autorizar a retirada da bicicleta pelo usuário após
Temperatura	(a) (b) (c) (d)	pagamento do aluguel da bicicleta se ele estiver a no máximo 3 metros de distância do bicicletário.
Rotação	(a) (b) (c) (d)	(c)Informar dados de identificação da bicicleta que pode ser
Bateria	(a) (b) (c) (d)	retirada, hora da retirada e a quantidade de horas alugadas, ou seja, disponíveis para uso.
Outro Sensor	(a) (b) (c) (d)	(d)Não autorizar a retirada da bicicleta pelo usuário. Informar motivo da não autorização de retirada da bicicleta. (Motivos da não autorização: Estar a mais de 3 metros de um bicicletário e/ou pagamento não autorizado).

Cenário 2 - Consultar Aplicativ

Elementos	Requisitos do Elemento	Descrição da APP
Localização	(a) (b) (c) (d)	Enquanto o usuário estiver com a bicicleta, o aplicativo deve ter as seguintes informações:
Conexão	(a) (b) (c) (d)	(a)Quando o tempo restante para uso da bicicleta chegar a 10 minutos, o usuário deverá ser informado do tempo
Tempo	(a) (b) (c) (d)	restante para uso e uma opção para o aluguel de mais horas. (b) Se o usuário ultrapassar o tempo permitido para uso da
Temperatura	(a) (b) (c) (d)	bicicleta, o sistema de alarme é acionado. Para o usuário, deve ser informado o motivo do acionamento do alarme da
Rotação	(a) (b) (c) (d)	bicicleta e oferecida a possibilidade de alugar mais horas. (c)Informar a temperatura do ambiente quando: estiver Frio
Bateria	(a) (b) (c) (d)	(abaixo de 21 °C) e Quente (acima de 25°C). (d) Informar Estado da Bateria do Celular quando: Baixa
Outro Sensor	(a) (b) (c) (d)	(abaixo 21%) e Alta (entre 90% e 100%).

Cenário 3 - <u>Devolver Bicicleta</u>

Elementos	Requisitos do Elemento	Descrição da APP						
Localização	(a) (b) (c)	Ao devolver a bicicleta, o aplicativo deve ser capaz de se						
Conexão	(a) (b) (c)	comunicar com o ponto de bicicletas. Essa comunicação deve ser composta por:						
Tempo	(a) (b) (c)	(a)Identificar que a bicicleta está corretamente encaixada no bicicletário, travar bicicleta e informar ao usuário a hora da						
Temperatura	(a) (b) (c)	devolução e a quantidade de horas disponíveis para uso. (b) Identificar que a bicicleta não está corretame						
Rotação	(a) (b) (c)	encaixada e informar ao usuário. (c)A rotação de tela só é permitida na tela após pesquisa por						
Bateria	(a) (b) (c)	pontos de bicicleta.						
Outro Sensor	(a) (b) (c)							

3. Gere casos de teste para cada Cenário.

Apêndice F - Roteiro do Participante - 1° Sessão APP2 - iFOODTRUCK

Tarefas:

1. De acordo com o documento de requisitos da Aplicação 2 – iFOODTRUCK, identifique e marque com um X, os elementos de contexto existentes em cada cenário.

Cenários	Elementos							
1 Pesquisar Food Trucks	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor	
2 Selecionar Food Truck e o Prato Desejado	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor	
3 Acompanhar Pedido	Localização	Conexão	Tempo	Temperatura	Rotação	Bateria	Outro Sensor	

2. Verifique na descrição da app, quais os requisitos para cada elemento em cada cenário.

Cenário 1 - Pesquisar Food Trucks

Elementos	Requisitos do Elemento	Descrição da APP
Localização	(a) (b)	Ao abrir o aplicativo, deve ser exibida uma lista com
Conexão	(a) (b)	os food trucks existentes por ordem:
Tempo	(a) (b)	(a)Ordem de distância: (menor distância entre ponto
Temperatura	(a) (b)	de origem do usuário e o ponto do food truck, destino do
Rotação	(a) (b)	usuário)
Bateria	(a) (b)	(b)Ordem de acordo com o clima. Acima de 30 °C a
Outro Sensor	(a) (b)	ordem é de preferência para gelaterias, sorveterias ou
		especializadas em bebidas.

0	Calcaianan		Drata	Daggiada
Cenario 2 -	Selecionar	FOOG Truck	t e o Praio	Deseiado

Elementos	Requisitos do Elemento	Descrição da APP
Localização	(a) (b) (c)	(a)A rotação de tela é permitida somente nesta tela de
Conexão	(a) (b) (c)	selecionar pedido, logo esta tela pode ficar tanto na horizontal quanto na vertical.
Tempo	(a) (b) (c)	(b)Após o usuário selecionar o food truck e o prato que deseja, devem estar disponíveis informações de tempo de
Temperatura	(a) (b) (c)	duração de preparo do prato e tempo para o usuário chegar ao Foodtruck. Após realização de pagamento, o pedido é
Rotação	(a) (b) (c)	autorizado e o nome do usuário entra na fila de espera.
Bateria	(a) (b) (c)	(c)Após ultrapassar o horário de funcionamento, a opção de confirmar o pedido deve ser desabilitada e o motivo deve ser
Outro Sensor	(a) (b) (c)	informado.

Cenário 3 - Acompanhar Pedido

Elementos	Requisitos do Elemento	Descrição da APP
Localização	(a) (b) (c) (d)	Após o pedido ser realizado, uma opção de exibição de
Conexão	(a) (b) (c) (d)	câmera é oferecida.
Tempo	(a) (b) (c) (d)	(a)Se a internet for wi fi, ela exibirá ao vivo o preparo do
Temperatura	(a) (b) (c) (d)	pedido.
Rotação	(a) (b) (c) (d)	(b)Se a internet for rede de dados, ela apenas exibirá uma tela
Bateria	(a) (b) (c) (d)	com a receita e informará a necessidade de wi fi.
Outro Sensor	(a) (b) (c) (d)	(c)Se a bateria estiver abaixo de 15%, ela apenas exibirá uma
		tela com a receita e informará que o celular necessita ser
		carregado.
		(d)Se a bateria estiver acima ou igual a 15% e menor que
		40%, ela exibirá uma mensagem que o celular estará
		prestes a descarregar e que necessita ser carregado
		para que a exibição do preparo do pedido continue ao
		vivo

3. Gere casos de teste para cada Cenário.

Apêndice G - Documento para a especificação dos casos de teste AD HOC

Hora	a Geral de Início dos Testes:	_ Final:			
Tes	Testador:				
Hor	a de Início:	Final: Data:			
Арр	avaliada: APP 2	Cenário: 2 Pesquisar Food Trucks			
Nún	nero de casos e testes criado	s:			
N°	Ação	Restrição	Resultado esperado		
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

Apêndice H - Roteiro do Participante – 2° Sessão

1. Para cada Cenário, Gere casos de teste utilizando a ferramenta.
2. Cenário 1: Pesquisar Pontos de Bicicletas: Hora de Início Fim
i. Informe aqui o Número de casos de teste criados pela ferramenta:
ii. Informe aqui o Número de casos de teste selecionados por você:
iii. Salve o resultado em .pdf com o nome "minhabicicleta_cenario1.pdf".
b. Cenário 2: Consultar Aplicativo Hora de Início Fim
i. Informe aqui o Número de casos de teste criados pela ferramenta:
ii. Informe aqui o Número de casos de teste selecionados por você:
iii. Salve o resultado em .pdf com o nome "minhabicicleta_cenario2.pdf".
c. Cenário 3: <u>Devolver Bicicleta</u> Hora de Início Fim
i. Informe aqui o Número de casos de teste criados pela ferramenta:
ii. Informe aqui o Número de casos de teste selecionados por você:
iii. Salve o resultado em .pdf com o nome "minhabicicleta_cenario3.pdf".
3. Ao finalizar o teste, informe aqui com precisão o horário de término:
4. Envie os documentos salvos por email para rdgosanjos@icomp.ufam.edu.br com o assunto
"SEU NOME – Resultados – Minha Bicicleta"

Apêndice I - Avaliação Pós Experimento - Teste para AMSC

De acordo cm a utilização das abordagens Ad Hoc e MTContext, verifique se você concorda com as afirmações
abaixo. Caso você "discorde" das afirmações, comente como podemos melhorar.

F	9	^	a	la	c

- (1) Discordo Totalmente; (2) Discordo; (3) Não tenho opinião formada; (4) Concordo; (5) Concordo Totalmente
- 1. Os "Tipos de Elementos" e as "Variações de Contexto" estão descritas de forma clara e objetiva na ferramenta.
- () Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente Se houver observações relacionadas a pergunta 1, utilize este espaço.

A navegação dentro da ferramenta é feita de forma fácil e intuitiva.

() Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente Se houver observações relacionadas a afirmação 2, utilize este espaço.

A tela de resumo "Test Context - Resume" contém todas as informações cadastradas até ali de forma ordenada e fácil para visualização do usuário.

- () Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente
- Se houver observações relacionadas a afirmação 3, utilize este espaço.

As sequências de teste geradas pela ferramenta estão exibidas de forma organizada e de fácil manuseio para cadastro dos campos "Behavior" e "Screenshots".

() Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente

Se houver observações relacionadas a afirmação 4, utilize este espaço.

- A "Dashboard" da ferramenta possui as informações necessárias para um caso de testes de aplicações sensíveis ao contexto. Logo, não há a necessidade de exclusão ou inclusão de alguma outra informação na "Dashboard".
- () Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente

Se houver observações relacionadas a afirmação 5, utilize este espaço.

Todas as informações contidas na "folha de impressão" da "Dashboard" são importantes. Sendo assim, não há a necessidade de exclusão ou inclusão de alguma outra informação na folha de impressão.

() Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente

Se houver observações relacionadas a afirmação 6, utilize este espaço._____

- 7. É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem MTContext que normalmente não é possível encontrar apenas com a abordagem Adhoc.
- () Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente

Se houver observações relacionadas a afirmação 7, utilize este espaço._____

- 8. É possível encontrar casos de teste para aplicações móveis sensíveis ao contexto com a abordagem Adhoc que não é possível encontrar apenas com a abordagem MTContext.
- () Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente

Se houver observações relacionadas a afirmação 8, utilize este espaço._____

9. Você utilizaria futuramente a abordagem MTContext para geração de casos de testes para aplicações
móveis sensíveis ao contexto.
() Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente
Se houver observações relacionadas a afirmação 9, utilize este espaço
10. A abordagem MTContext gera casos de teste demais e muitos desses casos de teste são
desnecessários para aplicações móveis sensíveis ao contexto.
() Discordo Totalmente; () Discordo; () Não tenho opinião formada; () Concordo; () Concordo Totalmente
Se houver observações relacionadas a afirmação 10, utilize este espaço.
Se houver qualquer outra observação ou sugestão de melhoria, descreva aqui